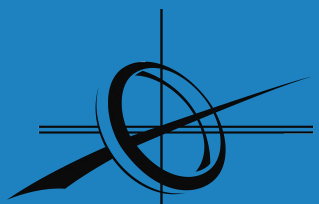
POLITÉCNICA



Universidad Politécnica de Madrid

ETSI SISTEMAS INFORMÁTICOS

Identificación de patrones de código HTML a través de modelos LLM

Proyecto Fin de Grado

Grado en Ingeniería del Software

Autor:
Jorge Daniel Cantuña Cabezas

Tutores:
Ángel Panizo Lledot

15/07/2025

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

Identificación de patrones de código HTML a través de modelos LLM

Proyecto Fin de Grado

Grado en Ingeniería del Software

Curso académico 2024-2025

Autor:
Jorge Daniel Cantuña Cabezas

Tutores:
Ángel Panizo Lledot

Quiero agradecer mucho a mi padre y a mi madre que siempre me motivan y me levantan cada vez que me caigo. Son el pilar de mi vida. Gracias por criarme con sus valores de trabajo y esfuerzo.

A mis hermanos que siempre me prestan su mano cuando lo necesito.

A mis amigos del colegio con los que llevo toda la vida junto y ya son parte de la familia.

A toda la gente que conocí en suecia y que hizo de mi estacia un momento maravilloso.

A Lali, por siempre sacarme una sonrisa, a Laura, por motivarme, y a todo el equipo Asento que formó gran parte de mis días.

A Miguel Ángel, que me aguanta y me empuja siempre para seguir adelante.

A todos los profesores y amigos que han hecho que disfrute de esta época universitaria.

Y sobre todas las cosas, agradecer a mi familia ecuatoriana que aunque no hablemos mucho y no nos veamos apenas, siempre les tengo en mente. Vayan bien o mal las cosas sé que ahí tendré mi verdadero hogar.

Gracias.

Resumen

No existe una estandarización en cuanto a la estructura HTML (HyperText Markup Language) de los periódicos digitales, es por eso que los procesos de extracción de datos, conocidos como *web scraping*, requieren una especialización para cada sitio web. Esta heterogeneidad, sumada a la constante evolución en el código de los medios digitales, hace que, actualmente, el web scraping sea un proceso específico de cada web y que requiera una adaptación continua.

Una posible solución a estos problemas de robustez y mantenibilidad de los procesos de web scraping tradicional son los modelos de lenguaje a gran escala, conocidos por sus siglas en inglés LLM (Large Language Models). Los LLM son modelos de inteligencia artificial entrenados con enormes cantidades de diferentes datos y textos, lo que les permite comprender patrones lingüísticos complejos y generar texto humano con mucha exactitud.

En este contexto, se pretende utilizar la capacidad de comprensión contextual de los LLM para abstraer el proceso de una estructura HTML particular, haciéndolo más adaptable a modificaciones y con menor necesidad de mantenimiento. Este trabajo busca intentar unificar el proceso de recolección automática de datos y, para ello, se ha desarrollado una aplicación web, denominada *NewScrAIper*. La aplicación constituye un sistema eficiente y versátil para la recopilación periódica de datos de noticias de periódicos digitales, empleando modelos LLM como herramienta clave para desvincular el proceso del medio.

NewScrAIper permite al usuario establecer y ejecutar procesos de web scraping de manera periódica, seleccionando el medio digital del cual desea extraer información, estableciendo palabras clave para filtrar los contenidos de su interés y definiendo un rango específico de fechas. La aplicación obtiene información relevante de cada noticia, que incluye el titular, el autor, la fecha de publicación, el subtítulo o un pequeño resumen y el enlace a la noticia completa.

Para lograr esta funcionalidad, la aplicación hace uso de un modelo LLM para la identificación y extracción de los datos de noticias de cualquier medio digital, permitiendo así tener un proceso generalizado e independiente del periódico digital. Esta información se almacena en una base de datos, en un formato semi-estructurado, para su posterior consulta y análisis.

Además, se ha implementado una interfaz de usuario con un diseño atractivo y sencillo, que garantiza una experiencia accesible y facilita la creación y monitorización de los procesos de extracción de datos. La aplicación puede ser probada siguiendo las instrucciones disponibles en el siguiente repositorio:
github.com/ec-dani/newscraiper.

Abstract

There is no standardization regarding the HTML (HyperText Markup Language) structure of digital newspapers, which is why data extraction processes, known as web scraping, require specialization for each website. This heterogeneity and the constant evolution in the code of digital media outlets means that, currently, web scraping is a process specific to each website and requires continuous adaptation.

One possible solution to these problems of robustness and maintainability in traditional web scraping processes is Large Language Models (LLMs). LLMs are artificial intelligence models trained with enormous amounts of different data and texts, allowing them to understand complex linguistic patterns and generate human-like text with great accuracy.

In this context, the aim is to use the contextual understanding capacity of LLMs to abstract the process from a specific HTML structure, making it more adaptable to modifications and requiring less maintenance. This project aims to unify the automatic data collection process, and for this purpose, a web application named *NewScrAIper* has been developed. The application serves as an efficient and flexible system for regularly gathering news data from online newspapers, employing LLM models as a key tool to separate the process from the specific media outlet.

NewScrAIper allows the user to set up and execute web scraping processes periodically, selecting the digital media outlet from which they want to extract information, establishing keywords to filter content of interest, and defining a specific date range. The application obtains relevant information from each news article, which includes the headline, author, publication date, subtitle or lead, and the link to the original article.

To achieve this functionality, the application uses an LLM model for the identification and extraction of news data from any digital media outlet, enabling a generalized process independent of the digital newspaper. This information is stored in a database, in a semi-structured format, for subsequent querying and analysis.

Additionally, a user interface with an appealing and user-friendly design has been implemented, providing an accessible experience and simplifying the creation and monitoring of data extraction processes. The application can be tested by following the instructions available in this repository:github.com/ec-dani/newscraiper.

Índice

Agradecimientos I
Resumen II
Abstract III

1. Introducción **1**
1.1. Contexto 1
1.2. Objetivos 2

2. Estado del arte **4**
2.1. Base de datos no relacionales 4
2.2. REST API 5
2.3. Arquitectura de microservicios 6
2.4. Sistema de gestión de tareas 7
2.5. Frontend 8
2.6. Herramientas tradicionales de *web scraping* 8
2.6.1. Beautiful Soup 9
2.6.2. Selenium 9
2.6.3. Scrapy 9
2.7. Large Language Model 10
2.7.1. Funcionamiento de los LLM 11
2.7.2. Cómo se entrenan y utilizan los LLM 12

3. Desarrollo del proyecto **14**
3.1. Herramientas de *NewScrAIper* 14
3.1.1. Langchain 14
3.1.2. FastAPI 14
3.1.3. Streamlit 15
3.1.4. Celery 15
3.1.5. RedBeat 16
3.1.6. Redis 16
3.1.7. MongoDB 16
3.1.8. Crawl4AI 17
3.1.9. Docker 17
3.2. Metodología 17
3.3. Estructura Microservicios 19
3.3.1. Contenedor Frontend 20
3.3.2. Contenedor Backend 20
3.3.3. Contenedor de RedBeat 22
3.3.4. Contenedor de Celery 22
3.3.5. Contenedor de Redis 25
3.3.6. Contenedor de MongoDB 25

4. Resultados **26**
4.1. Interfaz de Usuario 26
4.1.1. Create Task 27
4.1.2. List Expired Tasks 29
4.1.3. List Scheduled Tasks 30

4.1.4. Get Task Data 32
4.2. Casos de Uso 35
4.3. Resultados obtenidos 37
4.3.1. El País 37
4.3.2. La Razón 45
4.3.3. El Mundo 51
4.3.4. Explicación 59
4.3.5. ABC 60
4.4. Objetivos logrados 66
4.5. Problemas encontrados 67

5. Conclusiones y trabajos futuros **69**
5.1. Conclusiones 69
5.2. Lineas futuras 70
5.3. Impacto social y medioambiental 70

Bibliografía **72**

Anexos **73**

Índice de tablas

3.1. Endpoints FastAPI 21

Índice de figuras

3.1. *Diagrama Despliegue Contenedores. Los «node» representan contenedores Docker individuales. Las aristas representan las comunicaciones entre contenedores. Los «container» indican el tipo de componente software que se ejecuta dentro de cada contenedor* 20

4.1. *Interfaz Pantalla Principal* 26
4.2. *Interfaz Seleccionar Periódico* 27
4.3. *Interfaz Seleccionar Fechas* 28
4.4. *Interfaz Tarea Programada* 29
4.5. *Interfaz Tareas Expiradas* 30
4.6. *Interfaz Lista Tareas Programadas* 31
4.7. Interfaz Tarea Cancelada 32
4.8. Interfaz Obtener Datos 33
4.9. Interfaz Obtener Datos 34
4.10. Interfaz Obtener Datos Toolkit Tabla 34
4.11. Diagrama Casos de Uso 35
4.12. *Diagrama Secuencial Programar Tarea* 36
4.13. Diagrama Secuencial Extracción de Datos 36
4.14. El País 07/07/2025 37
4.15. El País 07/07/2025 38
4.16. El País 07/07/2025 38
4.17. La Razón 04/07/2025 46
4.18. La Razón 04/07/2025 47
4.19. El Mundo 17/06/2025 52
4.20. El Mundo 17/06/2025 52
4.21. El Mundo 17/06/2025 53
4.22. El Mundo 17/06/2025 53
4.23. ABC 04/07/2025 61
4.24. ABC 04/07/2025 62

Capítulo 1

Introducción

1.1. Contexto

En la era digital actual, la disponibilidad de información en la *World Wide Web* es inmensa tanto en volumen como en diversidad de formas de presentación [1], por ello un proceso genérico de extracción de datos, o *web scraping* en inglés, sería una herramienta de gran valor. Es posible que exista cierta similitud en la estructura a nivel visual, como puede ser el caso de los periódicos digitales. Sin embargo, si analizamos más a fondo, encontraremos que el código de cada medio digital es diferente. Esta heterogeneidad en el código fuente no solo se manifiesta entre distintos periódicos, sino que también es inherente a un mismo medio, el cual puede experimentar cambios estructurales de manera constante. Además, esta inestabilidad estructural se ve aún más afectada con la incorporación de elementos dinámicos [1]. Esto es gracias a AJAX (Asynchronous JavaScript and XML), un conjunto de técnicas que permiten actualizar el contenido sin recargar la página completa. Estas premisas hacen que el proceso tradicional de web scraping requiera una mejora encaminada hacia la adaptabilidad. La problemática radica en que cada medio digital posee una estructura HTML y un diseño únicos, lo que obliga a ajustar el proceso de scraping específicamente para cada sitio web. Esto se traduce en un aumento de costos en términos de tiempo, esfuerzo y mantenimiento. Por ello, se plantea la necesidad de avanzar hacia un enfoque más adaptable y robusto, capaz de generalizar el proceso.

El proceso de extracción de datos o web scraping sigue un esquema básico donde:

1. Se inicia una petición HTTP al servidor para obtener el contenido de la página
2. Se analiza el código HTML y CSS con el objetivo de indentificar y extraer la información deseada.
3. Finalmente, esta información extraida se guarda en una base de datos.

Es en la fase de análisis e identificación del contenido donde las herramientas tradicionales enfrentan mayores limitaciones. Actualmente, estas herramientas, como Beautiful Soup[1], dependen de patrones estáticos, como selectores de HTML, por lo que se ven superadas por las diversas formas en las que se estructuran las diferentes páginas web. Esto hace que al mínimo cambio en la disposición del código el proceso quede inservible. A esto se le suma la naturaleza dinámica de las webs modernas que provoca que su propia estructura se vea modificada con frecuencia, ya sea para hacerla visualmente más atractiva o porque así se definió su estructura. A día de hoy, tienes que navegar sobre el código HTML

[1] https://pypi.org/project/beautifulsoup4/

hasta encontrar el patrón que te lleve a la información deseada. Esto resulta en la obsolescencia de los *web scrapers*, ya que requieren una personalización específica para cada medio y un mantenimiento continuo.

En este contexto, los modelos LLM (Large Language Model) nos son de gran ayuda ya que son un tipo de modelo de procesamiento de lenguaje natural, o NLP (Natural Language Processing) en inglés. Por lo que tienen capacidad de entender, interpretar y generar lenguaje humano [2]. Estos modelos pueden analizar texto, realizar análisis de sentimiento y extraer información incluso de documentos con formatos inconsistentes [3]. Así como entienden texto, también han mostrado tener buena capacidad para interpretar código [4] lo que les dota de la adaptabilidad de la que carecen las actuales herramientas de scraping. Por estos motivos, será la pieza clave de nuestra aplicación *NewScrAIper*[2].

1.2. Objetivos

- **Objetivo General:** Diseñar y desarrollar un aplicación web open-source que constituya un sistema general y automático de extracción de noticias de portada de los principales periódicos digitales españoles mediante el uso de herramientas de inteligencia artificial y un flitrado por keywords, para su posterior análisis. Los datos que se extraerán son: título, autor, fecha, subtitulo o lead y el enlace a la noticia original. Se realizará un piloto, o llamado también por su siglas en ingles MVP (Minimum Valuable Product), en el que nos centraremos en 4 de los principales periódicos españoles: El País, El Mundo, ABC y La Razón.
- **Objetivos específicos:**
  1. **Construir un módulo de web scraping adaptativo**: Diseñar un componente o una serie de de componentes que integren la lógica necesaria para poder llevar a cabo procesos de extracción de datos que no requieran una especialización en una web en concreto, si no que sea un proceso general para todos los periódicos digitales.
  2. **Desarrollar un sistema de gestión de tareas:** Diseñar un sistema eficiente que permita programar los procesos de web scraping para que se ejecuten automaticamente en segundo plano.
  3. **Implementar una interfaz atractiva:** Construir un frontend que sea atractivo y sencillo de entender para que el usuario pueda navegar e interactuar con todas las funcionalidades de la aplicación.
  4. **Implementar una REST API consistente:** Desarrollar un backend que permita el intercambio de información desde el frontend con todo el resto de la lógica del sistema.
  5. **Crear una base de datos no relacional:** Diseñar una base de datos basada en collecciones que permita almacenar gran cantidad de información. En concreto, cada elemento de la colección representará los datos extraidos de los artículos de las noticias de portada.

[2] github.com/ec-dani/newscraiper

6. **Diseñar una arquitectura de miscroservicios:** Implementar una red de contenedores donde cada uno albergue un servicio necesario para el funcionamiento de todo el sistema.
7. **Implementar un ensayo con medios específicos:** Probar la aplicación extrayendo y procesando noticias de portada de El País, El Mundo, ABC y La Razón para validar la funcionalidad del sistema en un escenario real.

Capítulo 2

Estado del arte

En este apartado hablaremos del estado del arte de las principales tecnologías que se han mencionado en los objetivos y que han permitido el desarrollo de la aplicación *NewScrAIper*. Primero, hablaremos de las tecnologías relacionadas con el desarrollo de la aplicación, como son: bases de datos no relacionales, REST API, arquitectura de microservicios, sistema de gestión de tareas y frontend. Luego, introduciremos las herramientas actuales de web scraping. Por último, hablaremos de los modelos de lenguaje a gran escala.

2.1. Base de datos no relacionales

Las bases de datos no relacionales, también conocidas como NoSQL (Not only Structured Query Language), han ganado gran popularidad en los últimos años debido a su capacidad para manejar grandes volúmenes de datos no estructurados o semi-estructurados. A diferencia de las bases de datos relacionales, que se basan en esquemas fijos y el lenguaje SQL (Structured Query Language), las bases de datos NoSQL no requieren un esquema predefinido, lo que les otorga una gran flexibilidad.

Por un lado, las bases de datos SQL se basan en un modelo relacional, donde los datos se organizan en tablas con filas y columnas. Cada tabla tiene un esquema estricto, y las relaciones entre tablas se establecen mediante claves. Entre sus ventajas destacan la alta integridad de datos, la capacidad de realizar consultas complejas con SQL, y un gran respaldo por una amplia comunidad. Sin embargo, presentan limitaciones como la rigidez de su esquema, su poca eficiencia al manejar datos no estructurados y dificultades en cuanto a escalabilidad y rendimiento con datos exageradamente grandes.

Por otro lado, las bases de datos NoSQL están diseñadas para manejar grandes volúmenes de datos no estructurados y distribuidos. No requieren un esquema fijo, lo que permite una mayor flexibilidad y agilidad en el desarrollo. Destacan por su escalabilidad horizontal y su eficiencia en operaciones de lectura/escritura en grandes volúmenes de datos. Esta característica las hace ideales para escenarios donde el modelo de datos es dinámico, evoluciona rápidamente o no encaja fácilmente en un formato estricto. Son particularmente eficientes en entornos distribuidos y escenarios de Big Data. Entre los inconvenientes se destaca una baja robustez en la integridad de los datos y limitaciones al ejecutar consultas complejas con múltiples relaciones.

Existen varios tipos, cada uno mejor dependiendo del caso de uso:

- **Documentales**: Almacenan datos en formato de documentos, como JSON o BSON. Se utilizan en aplicaciones donde la estructura de los datos no es

fija, como en catálogos de productos e-commerce, datos de sensores o para la gestión de contenido que incluyen artículos y vídeos. Algunos ejemplos son MongoDB[1] y Couchbase[2].

- **Clave-Valor**: Almacenan datos como una colección de pares clave-valor. La clave es un identificador único y el valor puede ser cualquier tipo de dato. Algunos de los casos de uso son en caches de aplicaciones para almacenar datos a los que se accede con frecuencia, datos de sesión de los usuarios en aplicaciones web o tablas de búsqueda . Algunos ejemplos son Redis[3] o Amazon DynamoDB[4].

- **Grafos**: Se utilizan para modelar y almacenar datos con relaciones complejas, donde los datos se representan como nodos y las conexiones como aristas. Son ideales para escenarios donde las relaciones entre los datos son tan importantes como los datos en sí mismos. Se usan en redes sociales, sistemas de recomendaciones o en el internet de las cosas. Entre los ejemplos más conocidos está Neo4j[5].

2.2. REST API

Representational State Transfer (REST) es un estilo arquitectónico que define reglas para el diseño de aplicaciones en red. Application Programming Interface (API) es un conjunto de reglas y protocolos de diseño de aplicaciones que permiten la conexión entre diferentes servicios. El término API REST hace referencia a un sistema que permite el intercambio de información entre cliente y servidor usando el protocolo HTTP (Hypertext Transfer Protocol).

Una API RESTful se basa en varios principios clave que garantizan su escalabilidad, simplicidad y eficiencia. Entre ellos se encuentra:

- **Separación entre cliente y servidor**: Lo que permite que ambos evolucionen de forma independiente

- **Diseño sin estado (stateless)**:El servidor no almacena ningún estado de la sesión del cliente, cada solicitud del cliente al servidor debe contener toda la información necesaria.

- **Interfaz uniforme**: Los recursos se identifican mediante URLs (Uniform Resource Locator). Los clientes interactúan con los recursos obteniendo una representación"de los mismos, como por ejemplo un JSON, y las respuestas deben incluir información suficiente para ser interpretadas por sí solas.

Las operaciones más comunes en una API REST se mapean a los métodos HTTP estándar:

- **GET**: Para recuperar recursos.

[1] https://www.mongodb.com/
[2] https://www.couchbase.com/
[3] https://redis.io/
[4] https://aws.amazon.com/es/dynamodb/
[5] https://neo4j.com/

- **POST**: Crear un nuevo recurso.
- **PUT**: Actualizar un recurso existente (reemplazándolo completamente).
- **PATCH**: Actualizar parcialmente un recurso existente.
- **DELETE**: Eliminar un recurso.

Flask[6], FastAPI[7] o Django[8] son algunos de los frameworks más populares en Python que nos permiten construir una API RESTful de manera sencilla.

Aunque REST sigue siendo muy utilizado y es el estándar en muchas aplicaciones web, existen alternativas más populares para casos específicos. Una de ellas es **GraphQL**[9], un lenguaje de consulta para API que permite a los clientes pedir solo los datos que necesitan, evitando transferencias innecesarias. Otras herramientas como **WebSockets**, que permiten una comunicación bidireccional en tiempo real, o **Server-Sent Events**, que permiten una comunicación unidireccional servidor-cliente en tiempo real, complementan a REST. Por el contrario, tecnologías más antiguas como **SOAP**(Simple Object Access Protocol) han caído en desuso por su complejidad, mientras que **XML-RPC** (Extensible Markup Language Remote Procedure Call) y **JSON-RPC** han disminuido su uso a casos puntuales.

2.3. Arquitectura de microservicios

La arquitectura de microservicios es un enfoque de desarrollo de software que estructura una aplicación como una colección de servicios pequeños, independientes y débilmente acoplados, en lugar de un único bloque monolítico. Al contrario que en un sistema monolítico, donde todas las funciones de la aplicación están estrechamente unidas en un mismo bloque, en microservicios cada uno de estos servicios está separado en varios bloques y cada bloque está diseñado para realizar una función específica y bien definida.

Las características fundamentales de esta arquitectura son:

- **Servicios Independientes:** Cada microservicio es una unidad autónoma que puede ser desarrollada, desplegada y escalada de forma independiente del resto.
- **Comunicación Ligera:** Los microservicios se comunican entre sí a través de protocolos ligeros, como el protocolo HTTP.
- **Tecnologías Heterogéneas:** Los microservicios permiten el uso de diferentes tecnologías, lenguajes de programación y frameworks dentro de la misma aplicación global.
- **Despliegue Independiente:** La capacidad de desplegar cada servicio de forma aislada permite un ciclo de desarrollo y despliegue continuo más rápido y con menos riesgos. Si un cambio en un microservicio falla, solo afecta a ese servicio y no a toda la aplicación.

[6]https://flask.palletsprojects.com/en/stable/
[7]https://fastapi.tiangolo.com/
[8]https://www.djangoproject.com/
[9]https://graphql.org/

- **Escalabilidad Horizontal:** Los microservicios facilitan la escalabilidad. Si un servicio en particular experimenta una alta demanda, se pueden lanzar múltiples instancias de ese servicio de forma independiente, sin necesidad de escalar toda la aplicación.
- **Tolerancia a Fallos:** Un fallo en un microservicio no necesariamente derriba toda la aplicación, ya que solo.

Para el despliegue, orquestación y escalado automatizado de estos servicios, las herramientas más utilizadas son Docker y Kubernetes.

2.4. Sistema de gestión de tareas

Para manejar procesos recurrentes o que llevan mucho tiempo, se emplean sistemas de tareas en segundo plano. Estos sistemas nos sirven para desacoplar operaciones que consumen muchos recursos de la interacción directa con el usuario.

Estos tipos de sistemas funcionan mediante un intermediario, conocido como *broker*, de mensajes que recibe y almacena las tareas en una cola. Luego, trabajadores, o *workers* (procesos independientes), toman esas tareas y las ejecutan de manera asíncrona. Esto permite que el sistema que inicia la tarea, como una aplicación web, responda al usuario de inmediato sin esperar a que la operación finalice, logrando así un fuerte desacoplamiento entre la solicitud y la ejecución.

Beneficios clave de un sistema de gestión de tareas:

- **Mejora la experiencia del usuario:** La aplicación responde más rápidamente al usuario, ya que las operaciones pesadas se mueven a segundo plano.
- **Escalabilidad:** Se pueden añadir más trabajadores según sea necesario para manejar un mayor volumen de tareas, escalando la capacidad de procesamiento sin afectar la interfaz de usuario.
- **Tolerancia a Fallos:** Si un trabajador falla, la tarea puede ser reintentada por otro trabajador. Además, el broker de mensajes puede persistir las tareas, asegurando que no se pierdan incluso si el sistema falla.
- **Gestión de Carga:** Permite distribuir la carga de trabajo entre múltiples trabajadores, optimizando el uso de recursos.

Herramientas como Celery[10] (en combinación con Redis o RabbitMQ como broker), Apache Kafka[11], Amazon SQS[12] o Apache ActiveMQ[13] permiten gestionar colas de tareas de forma eficiente.

[10] https://docs.celeryq.dev/en/stable/
[11] https://kafka.apache.org/
[12] https://aws.amazon.com/es/sqs/
[13] https://activemq.apache.org/

2.5. Frontend

El frontend es la parte de la aplicación que interactúa directamente con el usuario en el navegador o dispositivo móvil. Se encarga de la interfaz de usuario (UI) y experiencia del usuario (UX), buscando que la aplicación sea fácil de usar, estéticamente agradable y que funcione en distintos dispositivos. Sus componentes básicos son:

- **HTML (HyperText Markup Language):** Es el lenguaje de marcado estándar para crear páginas web. Define la estructura y el contenido de una página web.
- **CSS (Cascading Style Sheets):** Es un lenguaje de hoja de estilos que permite controlar el diseño, los colores, las fuentes, los espacios, los efectos visuales y la disposición de los elementos en la página.
- **JavaScript (JS):** Es un lenguaje de programación fundamental para el desarrollo web. Permite crear interactividad dinámica en las páginas web. Con JavaScript, el frontend puede responder a las acciones del usuario (clics, movimientos del ratón), validar formularios, cargar contenido de forma asíncrona (AJAX), crear animaciones, y en general, hacer que la página sea mucho más que un documento estático.

El desarrollo de frontend se ha estandarizado en frameworks y librerías como React, Angular, Vue o Streamlit. Estas herramientas permiten construir interfaces reactivas, modulares y mantenibles, centradas en la experiencia del usuario.

- **React**[14]**:** Una librería de javascript basada en componentes para construir interfaces de usuario. Es muy popular por su eficiencia y su capacidad para manejar aplicaciones de una sola página.
- **Angular**[15]**:** Un framework completo y robusto para construir aplicaciones web grandes y complejas. Sigue una arquitectura más prescriptiva y ofrece un ecosistema más extenso.
- **Vue.js**[16]**:** Un framework de jaavasriptfácil de aprender, ligero, muy versátil y que puede adaptarse a diferentes escalas de proyectos.
- **Streamlit**[17]**:** Un framework open-source ligero y sencillo para crear aplicaciones de datos interactivas. Ideal para prototipado rápido, visualización de datos, machine learning o creación de dashboards, todo con muy poca configuración y un enfoque práctico centrado en Python.

2.6. Herramientas tradicionales de *web scraping*

Cuando hablamos de herramientas tradicionales en este contexto, estamos hablando de aquellas que se basan en instrucciones específicas y reglas fijas, que

[14] https://es.react.dev/
[15] https://angular.dev/
[16] https://vuejs.org/
[17] https://streamlit.io/

usan selectores estáticos como XPath (un lenguaje para navegar y seleccionar nodos en documentos XML o HTML) o CSS para extraer datos. Estas herramientas recorren la estructura HTML en busca de elementos con etiquetas concretas. Estas herramientas tradicionales muestran dos grandes limitaciones: deben ser personalizadas para cada página web y no están adaptadas para tratar elementos dinámicos. Para ese caso se han seleccionado tres librerías Python muy representativas y utilizadas en el web scraping: Beautiful Soup, Selenium y Scrapy. Estas herramientas ilustran el estado del arte del web scraping tradicional, mostrando las capacidades y limitaciones de estos procesos.

2.6.1. Beautiful Soup

Beautiful Soup[18] es una biblioteca de Python que convierte el código fuente HTML crudo en una estructura de datos fácil de usar. Obtiene el código de la página y lo estructura como un árbol de tal manera que cada elemento tiene padre, hijo y/o hermano. Normalmente, se utiliza para extraer información de páginas web estáticas, donde el contenido está completamente cargado en el HTML inicial. Es muy útil si deseas extraer algún texto, enlace o imagen que sea visible en el código fuente.

La fortaleza de esta librería es su simpleza, ya que su sintaxis es clara y fácil de entender, y su compatibilidad con otras herramientas de Python. Por el contrario, sus principales desventajas surgen cuando encuentras contenido dinámico, como por ejemplo contenido que es renderizado por JavaScript. Beautiful Soup solo mira el código HTML, por lo que estos elementos serían ignorados.

2.6.2. Selenium

Selenium[19] es un framework que controla navegadores web como Chrome, Firefox o Safari. No solo lee el HTML, ejecuta todo el JavaScript, CSS, y hace todo lo que haría un usuario real como si estuviera usando el sitio. Se utiliza cuando necesitas extraer datos de páginas web dinámicas, específicamente páginas que requieren acción del usuario para mostrar contenido.

Sus ventajas son el hecho de que puede manejar cualquier tipo de sitio web, sin importar cuán complejo sea el JavaScript. Es capaz de emular un comportamiento humano extremadamente realista, como retrasos aleatorios o movimientos del ratón. Sus desventajas son casi todas basadas en el rendimiento y los recursos. Al usar un navegador completo, se vuelve lento y consume mucha memoria, lo que lo hace excesivo para trabajos simples. Sin embargo, sigue siendo vulnerable a cambios estructurales en el código HTML, teniendo así que actualizarse cada vez que los sitios web modifican su arquitectura de elementos.

2.6.3. Scrapy

Scrapy[20] es un framework open-source de Python desarrollado concretamente para el web scraping. Proporciona casi todas las herramientas de programación

[18] https://pypi.org/project/beautifulsoup4/
[19] https://pypi.org/project/selenium/
[20] https://www.scrapy.org/

que necesitamos, todo el almacenamiento y procesamiento de datos que precisamos. Es útil en grandes proyectos donde hay que extraer miles o cientos de miles de páginas.

Scrapy destaca por su capacidad para procesar múltiples páginas de forma simultánea, por su robusto manejo de errores y reintentos, y por ofrecer funcionalidades muy útiles para la organización de los datos extraídos. Sin embargo, su principal limitación radica en la dificultad para trabajar con contenido dinámico. Dado que utiliza selectores estáticos, como XPath o CSS, puede no ser capaz de acceder a toda la información.

2.7. Large Language Model

Los modelos de lenguaje a gran escala o LLM (Large Language Models) son un tipo de modelo de inteligencia artificial que pertenece al campo del aprendizaje automático (machine learning). Más concretamente, a las redes neuronales profundas [5]. Su principal función es procesar, comprender y generar lenguaje humano de forma automática [5].

Estos modelos han sido entrenados con grandes cantidades de texto de diferentes fuentes, como libros, artículos o páginas web. Gracias a este entrenamiento, son capaces de realizar una amplia gama de tareas relacionadas con el procesamiento del lenguaje natural (NLP, por sus siglas en inglés), como responder preguntas, redactar textos, traducir idiomas o analizar sentimientos [6]. Actualmente, la mayoría de los LLM, y en los que nos vamos a centrar, están basados en la arquitectura de *transformers*[7]. Esta arquitectura permite procesar grandes secuencias de texto en paralelo mediante mecanismos de auto-atención (self-attention), lo que les da la capacidad de entender el contexto de las palabras de una forma muy eficiente.

Un aspecto clave de estos modelos es su tamaño, el cual se mide por el número de parámetros. Los parámetros son valores internos que el modelo ajusta durante el entrenamiento para aprender cómo transformar las entradas en salidas adecuadas. Modelos como GPT de OpenAI o LLaMa de Meta cuentan con decenas o incluso cientos de miles de millones de parámetros, lo que les confiere una gran capacidad para generar texto coherente y realizar tareas complejas con un rendimiento cercano al humano [5].

Es importante destacar que estos modelos mencionados, GPT y LLaMa, pertenecen a dos tipos diferentes de modelos: modelos privados y modelos open-source. Los modelos privados, como GPT, son desarrollados por una empresa que restringe el acceso al código y a los datos de entrenamiento. En cambio, los modelos open-source, como LLaMa, liberan el código, permitiendo que cualquier usuario tenga uso libre del modelo y lo modifique libremente, si lo desea. Los principales representantes del estado de ambos tipos son:

- **ChatGPT (OpenAI)** [21]**:** Un ejemplo clave de un modelo privado ofrecido a través de una API y una interfaz de chat web. Se ha convertido en

[21]https://openai.com/

uno de los modelos más conocidos para el público general. Basado en la arquitectura GPT (Generative Pre-trained Transformer), destaca por su capacidad para generar texto coherente y contextualmente relevante, responder preguntas, resumir información y participar en conversaciones fluidas.

- **Claude (Anthropic)**[22]**:** Desarrollado por Anthropic, Claude es otro LLM de alto rendimiento que compite directamente con los modelos GPT. Se distingue por su enfoque en la seguridad y la ÏA constitucional", lo que significa que ha sido entrenado siguiendo un conjunto de principios para ser más útil, inofensivo y honesto. Claude es particularmente valorado en entornos que requieren un alto nivel de fiabilidad y alineación. También es un modelo privado accesible a través de API y, recientemente, con una interfaz web .

- **LLaMA (Meta AI)**[23]**:** Publicado por Meta AI, LLaMA (Large Language Model Meta AI) ha tenido un impacto revolucionario en la comunidad de código abierto. Se han caracterizado por su gran rendimiento y su disponibilidad como modelos de código abierto, democratizando el acceso a este tipo de modelos. Esto ha permitido a investigadores y desarrolladores de todo el mundo construir sobre estos modelos, afinarlos para tareas específicas y desplegarlos en diversas aplicaciones sin las restricciones de los modelos propietarios.

- **Mistral AI (Mixtral)**[24]**:** Fundada por antiguos investigadores de Google DeepMind y Meta, Mistral AI ofrece un modelo código abierto muy capaz y eficiente. Mixtral 8x7B es un modelo cuya arquitectura de mezcla de expertos (Mixture-of-Experts, MoE), le permite lograr un rendimiento comparable al de modelos mucho más grandes con una eficiencia computacional superior. Esto lo hace muy atractivo para su despliegue y afinamiento, especialmente en entornos donde los recursos son una restricción importante.

- **Gemini (Google)** [25]**:** Un modelo privado desarrollado por Google, que pertenece a los modelos multimodales diseñado para ser altamente flexible y capaz de comprender y operar a través de diferentes tipos de información, incluyendo texto, imágenes, audio y video.Esta capacidad multimodal, le facilita el procesamiento y razonamiento sobre información que integra varias modalidades al mismo tiempo.Gemini se ofrece en varias versiones optimizadas para diferentes casos de uso, desde modelos ligeros para dispositivos móviles (Gemini Nano) hasta los más potentes para tareas complejas (Gemini Ultra). Google no ha liberado el código fuente completo de las versiones más avanzadas, pero sí ofrece acceso a través de APIs.

2.7.1. Funcionamiento de los LLM

La base fundamental de los LLM es la arquitectura basada en *transformers*,un tipo de red neuronal . Esta arquitectura elimina las capas recurrentes o convolucionales

[22]https://www.anthropic.com/
[23]https://www.llama.com/
[24]https://mistral.ai/
[25]https://deepmind.google/

de las redes neuronales y las sustituye por auto-atención multicabezal (multi-headed self-attention)[7].

El proceso de un transformer comienza dividiendo la secuencia en pequeñas unidades, llamadas tokens, que podrían ser palabras enteras o fragmentos. Estos tokens luego se convierten en vectores numéricos. Una vez representados numéricamente, el modelo aplica el mecanismo de autoatención (self-attention) para determinar la relevancia de cada token en relación con los demás dentro de la misma secuencia. Este cálculo se realiza de forma paralela en múltiples cabezales de atención (multi-head attention), lo que permite captar distintos matices de las relaciones contextuales entre las palabras [7].

Posteriormente, cada token, ahora enriquecido con nueva información, pasa por una red simple (feed-forward) para prepararse para la siguiente capa. Al repetir este proceso, el transformer adquiere una comprensión detallada del texto. Finalmente, usa estas representaciones internas para predecir el próximo token o proporcionar una respuesta [7].

Gracias a esta estructura, los transformadores escalan de manera eficiente. A medida que se incrementa el tamaño del modelo, la cantidad de datos y los recursos computacionales, mejora el rendimiento del sistema. Este fenómeno ha sido estudiado bajo el marco de las leyes de escalado (scaling laws), que explican cómo la ampliación coordinada de los componentes del modelo conduce a mejoras sistemáticas en su capacidad de generalización [8].

2.7.2. Cómo se entrenan y utilizan los LLM

El entrenamiento de un LLM consta de dos fases principales. Primero, se emplea el aprendizaje autorregresivo, en el que el modelo aprende a predecir el siguiente token en una secuencia a partir de los anteriores, de forma similar a un autocompletado de texto [7]. Esta fase inicial dota al modelo de una comprensión estadística de la estructura del lenguaje. Después, se aplica RLHF (Reinforcement Learning from Human Feedback), una técnica en la que los desarrolladores corrigen y valoran las respuestas del modelo para generar una señal de recompensa que guíe el ajuste fino de sus parámetros [8]. Esta técnica permite alinear las respuestas del modelo con las preferencias humanas, transformándolos en modelos asistentes capaces de sostener diálogos complejos [6].

En la práctica, los usuarios interactúan con los LLM a través de indicaciones textuales denominadas *prompts*, que funcionan como instrucciones o contextos sobre los que el modelo genera una respuesta. Este proceso, conocido como ingeniería de *prompts*, permite adaptar el modelo a una gran variedad de tareas sin necesidad de modificar su estructura interna [5]. Un parámetro crucial con el que el usuario puede interactuar es la temperatura (temperature). Este parámetro es un valor numérico que controla la aleatoriedad y creatividad de las respuestas del modelo [26].Una temperatura baja, cercana a 0.0, hace que el modelo seleccione la palabra con mayor probabilidad. Esto implica que el modelo se vuelve determinista,

[26]LangChain Model Temperature

preciso y repetitivo. Por el contrario, una alta temperatura , mayor que 1.0, hace que el modelo elija la palabra con menos probabilidades, volviendo al modelo más aleatorio y creativo, pero también genera respuestas más incoherentes.

Capítulo 3

Desarrollo del proyecto

En esta sección, explicaremos las herramientas, la metodología y la arquitectura utilizadas para desarrollar la aplicación **NewScrAIper**[1].

3.1. Herramientas de *NewScrAIper*

A continuación se detallan las herramientas que se han usado para el desarrollo de la aplicación.

3.1.1. Langchain

Lanchain[2] es un framework open-source diseñado para facilitar la construcción de aplicaciones que utilizan modelos de lenguaje a gran escala (LLMs) de manera eficiente y estructurada. Ofrece una gran variedad de funciones clave y componentes modularizados para integrar LLMs con diferentes fuentes de datos o APIs. Algunas de estas funciones incluyen memoria conversacional, cadenas de procesamiento (chains), agentes autónomos y conexión a bases de datos vectoriales. Gracias a estas funcionalidades, podemos crear chatbots inteligentes, sistemas de preguntas y respuestas sobre documentos, agentes de IA que pueden usar herramientas externas, aplicaciones RAG (Retrieval-Augmented Generation). Algunos puntos fuertes son:

- Su capacidad para integrar múltiples herramientas y fuentes de datos, lo que permite construir aplicaciones más robustas y con conocimiento contextualizado.
- La existencia de una interfaz estándar para cadenas (chains), agentes y memoria, facilitando la reutilización y combinación de componentes.
- Constituye un ecosistema muy completo con una gran comunidad que apoya y permite evolucionar esta herramienta.

En cuanto a puntos débiles, cabe destacar la juventud de la librería, ya que lleva poco tiempo a la luz, lo que hace que la documentación no sea exhaustiva. Al ser tan novedosa, también está en constante cambio, haciéndose difícil la curva de aprendizaje.

3.1.2. FastAPI

FastApi[3] es un framework web moderno y de alto rendimiento para construir APIs con Python. Algunas de las funcionalidades destacadas son la generación

[1] github.com/ec-dani/newscraiper
[2] https://www.langchain.com/
[3] https://fastapi.tiangolo.com/

automática de documentación, valida datos automáticamente y maneja de manera automática la conversión de objetos Python y JSON. Los puntos fuertes más destacables:

- **Rapidez**: Muy alto rendimiento, a la par con NodeJS y Go (gracias a Starlette y Pydantic). Uno de los frameworks de Python más rápidos disponibles.
- **Fácil**: Diseñado para ser fácil de usar y aprender. Menos tiempo leyendo documentación.
- **Corto**: Minimiza la duplicación de código. Múltiples funciones de cada declaración de parámetro. Menos errores.
- **Robusto**: Obtenga código listo para producción. Con documentación interactiva automática.
- **Basado en estándares**: Basado en los estándares abiertos para APIs: OpenAPI[4] y JSON Schema[5].

3.1.3. Streamlit

Streamlit[6] es un framework para crear aplicaciones web interactivas enfocadas a ciencia de datos y aprendizaje automático. Gracias a sus componentes de visualización de datos, hace de esta librería la herramienta perfecta para la creación de un frontend sencillo y funcional. Los puntos más destacables son:

- **Simple**: Escribe código bonito y fácil de leer.
- **Creación rápida** e interactiva de prototipos: Permita que otros interactúen con sus datos y proporcionen comentarios rápidamente
- **Edición en vivo**
- **Código abierto y gratuito**

3.1.4. Celery

Celery[7] es un potente sistema de colas de tareas distribuidas basado en Python y de código abierto que se utiliza para la ejecución asíncrona de tareas. Su objetivo es separar cualquier trabajo de larga duración de la línea principal de la aplicación, distribuyendo la carga de procesamiento entre múltiples trabajadores (workers). Se utiliza comúnmente para correos masivos, generación de informes, scraping web o cualquier cosa que pueda potencialmente bloquear tu aplicación.

Sus ventajas son la escalabilidad, tolerancia a fallos, el uso de varios brokers de mensajes y la posibilidad de programar tareas. En el lado negativo, introduce complejidad arquitectónica en el sistema, requiere un mayor nivel de configuración

[4]https://github.com/OAI/OpenAPI-Specification
[5]https://json-schema.org/
[6]https://streamlit.io/
[7]https://github.com/celery/celery

y monitoreo.

Celery tiene su propio programador de tareas, Celery Beat. Está diseñado para lanzar tareas de manera automática según un horario predefinido. Se suele usar para limpieza de datos, generación de reportes programados o sincronización de sistemas. Su mayor fortaleza es la buena integración con Celery. Una de sus grandes debilidades es que la definición de las tareas periódicas no se hace de manera dinámica, sino que deben estar definidas antes de arrancar el sistema.

3.1.5. RedBeat

Celery-RedBeat[8] es un planificador de tareas de Celery mejorado que utiliza Redis para guardar la información de las tareas. Anteriormente, cualquier cambio o actualización en una tarea requería que Beat, el planificador por defecto de Celery, se detuviera y se reiniciara. Sin embargo, ahora basta con actualizar una clave en Redis para que Beat detecte el cambio en el siguiente ciclo. Esta mejora permite programar las tareas de manera dinámica sin necesidad de reiniciar el sistema, perfecto para nuestro caso. Además, RedBeat ofrece una alta disponibilidad y escalabilidad, ya que al utilizar Redis como backend, múltiples instancias de Beat pueden leer y procesar las tareas programadas.

3.1.6. Redis

Redis[9] es una base de datos en memoria basada en estructuras de datos, de código abierto y con soporte para persistencia. Se utiliza frecuentemente como sistema de almacenamiento en caché, cola de mensajes o almacén de sesiones. Sus ventajas incluyen alta velocidad, soporte para estructuras de datos avanzadas y facilidad de uso. Como desventajas, su almacenamiento en memoria puede limitar la escalabilidad en grandes volúmenes de datos. En el contexto de Celery, Redis actúa como broker, es decir, actúa como un intermediario que recibe tareas de la aplicación y las distribuye a los trabajadores para su ejecución.Además, también puede actuar como backend para el almacenamiento de tareas y resultados.

3.1.7. MongoDB

Mongodb[10] es una base de datos NoSQL orientada a documentos. La base de datos NoSQL orientada a documentos más típica es MongoDB. Inserta documentos en un formato similar a JSON (BSON) para almacenar datos y está diseñada para ser una base de datos flexible y escalable. Se utiliza en aplicaciones web modernas, CMS, aplicaciones móviles, procesamiento de grandes volúmenes de datos y cuando la estructura de los datos está cambiando. Sus ventajas incluyen ser libre de esquemas, gran rendimiento de lectura, buena escalabilidad horizontal y facilita el desarrollo cuando trabajamos con objetos JSON. Estas características la hacen ideal ya que nuestros datos son semi-estructurados y tendremos cientos de documentos en una única colección.

[8] https://github.com/sibson/redbeat
[9] https://redis.io/es/
[10] https://www.mongodb.com/

3.1.8. Crawl4AI

Crawl4AI[11] es una herramienta innovadora de web scraping que integra técnicas convencionales de extracción de datos con avanzadas capacidades de inteligencia artificial. Su función principal es extraer y estructurar contenido web de forma inteligente, superando las limitaciones de los sitios web modernos que utilizan JavaScript y contenido dinámico. Esta herramienta actúa como un navegador altamente sofisticado, específicamente desarrollado para páginas web modernas. No solo se basa en selectores estáticos, integra diversas estrategias para capturar cada componente del sitio web. Permite utilizar un navegador basado en Chromium sin una interfaz gráfica, simulando que un humano está accediendo a la página. También es capaz de ejecutar JavaScript en tiempo real y realizar esperas inteligentes, esperando que sucedan cosas específicas. Además, puede simular el scroll y realizar clics en botones. Esto no solo permite obtener el HTML inicial del sitio, sino también los elementos dinámicos del sitio. Su gran punto en contra es su reciente creación, lo que hace que se estén descubriendo errores continuamente y cambiando los métodos, lo que empeora la curva de aprendizaje.

3.1.9. Docker

Docker[12] permite empaquetar una aplicación, incluyendo todas sus dependencias, bibliotecas y configuraciones, en un contenedor aislado que se ejecuta de manera consistente en cualquier sistema. Se basa en la virtualización a nivel de sistema operativo, pero a diferencia de una máquina virtual, no contiene todo el sistema operativo. Docker utiliza el núcleo del sistema anfitrión para crear los contenedores, haciéndolo de una manera más rápida y más eficiente en cuanto a recursos. Aprovecha el núcleo de Linux y su funcionalidad para segregar procesos y ejecutarlos de manera independiente. Es por eso que esta herramienta se ha convertido en una aliada clave de los microservicios, ya que permite empaquetar cada servicio en su propio contenedor, facilitando su independencia, escalabilidad y mantenimiento. Esta arquitectura se sustenta en tres componentes fundamentales: las imágenes, los contenedores y el Docker Engine.

3.2. Metodología

En el área de desarrollo de software, tener una metodología es muy importante. Elegir la metodología correcta facilita la organización, mejora la eficiencia del trabajo y aumenta la calidad del producto.

En este caso, nuestro objetivo es obtener un producto mínimo viable, conocido por sus siglas en inglés como MVP (Minimum Viable Product). Es decir, una aplicación con las funciones suficientes para que los primeros clientes puedan utilizarla y aportar sus sugerencias para futuros desarrollos. Además, muchas de las tecnologías usadas en este proyecto eran muy nuevas, así que el nivel de experiencia con estas era casi nulo. Teniendo así que dedicar parte del tiempo a aprender a usarlas. También cabe señalar que este producto no es un producto

[11] https://docs.crawl4ai.com/
[12] https://www.docker.com/

crítico, pues, si falla, los daños o pérdidas no serán graves.

Por estos motivos se ha decidido optar por las metodologías ágiles, centrándonos en construir un MVP a través de iteraciones del producto. Buscamos satisfacer los requisitos del cliente, que pueden cambiar, de la manera más rápida y efectiva. Los requisitos han sido escritos como historias de usuario, y son los siguientes[13]:

HU1. Como usuario, quiero seleccionar qué medios digitales scrapear, para obtener toda la información de las noticias de portada.

HU2. Como usuario, quiero introducir un rango de fechas, para definir el periodo de búsqueda de noticias.

HU3. Como usuario, quiero que el sistema obtenga periódicamente las noticias dentro del rango de fechas establecido, para asegurar la recopilación diaria de datos.

HU4. como usuario, quiero iniciar, monitorizar y cancelar un proceso de búsqueda de noticias, para tener control sobre el proceso de recopilación de noticias.

HU5. Como usuario, quiero especificar una o varias keywords, para filtrar las noticias.

HU6. Como usuario, quiero que la aplicación obtenga el titular de la noticia, para saber de qué trata la noticia.

HU7. Como usuario, quiero que a aplicación obtenga la fecha de la noticia, para saber cuándo fue publicada.

HU8. Como usuario, quiero que a aplicación obtenga el subtitulo o lead de la noticia, para saber los detalles de la noticia.

HU9. Como usuario, quiero que la aplicación almacene un enlace a la noticia original, para consultar la fuente completa si es necesario.

HU10. Como usuario, quiero que la aplicación almacene los datos de las noticias recolectadas en una base de datos, para su consulta y análisis.

HU11. Como usuario, quiero obtener las noticias almacenadas filtrándolas por fecha o keywords, para consultar y analizar rápidamente a la información que necesito.

HU12. Como usuario, quiero una interfaz funcional y fácil de navegar, para interactuar con todas las funcionalidades de la aplicación.

En este contexto, se ha decidido adoptar una metodología ágil basada en Scrum pero adaptada, centrada en desarrollo iterativo e incremental, buscando entregar el mayor valor al usuario.

En cuanto a la organización de Sprints, se ha decidido separar en intervalos de dos semanas, enfocando cada iteración en experimentar y/o desarrollar cada componente o grupos de componentes. La distribución que se seguirá es la siguiente:

[13]HU hace referencia Historia de Usuario

- **Sprint 1 (Spike Crawl4AI + LangChain)**:En este sprint, aprenderemos a utilizar las librerías LangChain y Crawl4AI. Crearemos la lógica mínima del LLM necesaria para la extracción del título, autor, fecha, URL original y lead de un solo artículo. Luego refinaremos este módulo para obtener los datos de noticias de diferentes periódicos. Las historias de usuario asociadas serán: **HU6**, **HU7**, **HU8**, **HU9**.
- **Sprint 3 (Spike Celery + RedBeat + Redis)**: Aquí experimentaremos con Celery, la configuración de RedBeat y monitorización. Estas librerias actúan como un gestor de tareas distribuido, lo que nos servirá para crear nuestro sistema de gestión de tareas. Las historias de usuario asociadas son: **HU3**, **HU4**
- **Sprint 4 (FastAPI + MongoDB)**: Diseñaremos el API REST que comunicará la inferfaz del usuario con toda la lógica de la aplicación. Las historias de usario relacionadas son:**HU1**, **HU2**, **HU4**, **HU10**.
- **Sprint 5 (Spike Streamlit)**: Nos familiarizaremos con la libreria Streamlit y crearemos el frontend para que el usuario interactúe de forma sencilla con la aplicación. Las historias de usario relacionadas son:**HU12**.
- **Sprint 6 (Spike Docker)**: Aprenderemos a usar Docker y trabajaremos en la arquitectura de microservicios conectando todos los componentes. Terminando así el MVP de *NewScrAIper*

3.3. Estructura Microservicios

La aplicación NewScrAIper está construida con un enfoque técnico de arquitectura de microservicios. Se utilizan varios contenedores Docker para encapsular los distintos servicios del sistema. A continuación, se proporciona una explicación de los diferentes contenedores que se pueden observar en la figura 3.1.

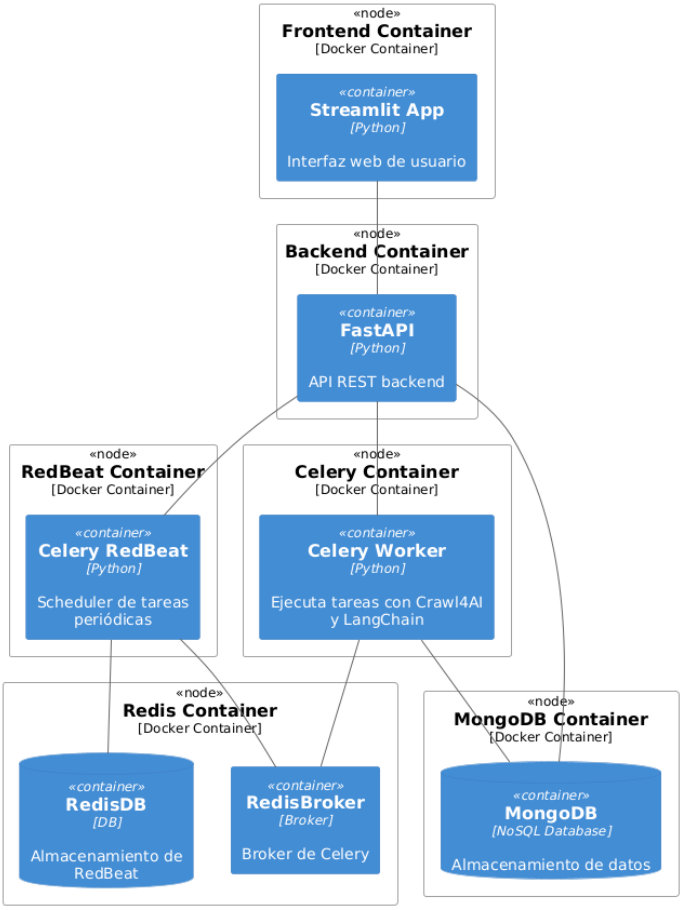


Figura 3.1: *Diagrama Despliegue Contenedores.*
Los «node» representan contenedores Docker individuales. Las aristas representan las comunicaciones entre contenedores. Los «container» indican el tipo de componente software que se ejecuta dentro de cada contenedor

3.3.1. Contenedor Frontend

El **contenedor Frontend** contiene la interfaz web que el usuario empleará para interactuar con el sistema. Para la creación de este frontend se ha usado Streamlit. Un framework de Python que nos permite crear una interfaz web de usuario amigable, sencilla y atractiva para nuestra aplicación.

3.3.2. Contenedor Backend

El **contenedor Backend** encapsula el servicio de la REST API que conecta el frontend con toda la demás lógica de la aplicación para que se pueda comunicar e intercambiar información de manera eficiente y rápida. Este contenedor se conecta directamente con el **contenedor de RedBeat**, para la programación de futuras tareas, el **contenedor de Celery**, para las tareas en ejecución y con

el **contenedor de MongoDB** de la base de datos, para obtener los resultados finales del proceso.

El **componente FastAPI** contiene toda la lógica de la REST API, para la que se ha usado el framework de FastAPI. Esta herramienta nos permite definir los endpoints necesarios para que se ejecute la lógica de la aplicación de una manera sencilla. Los endpoints están representados en la siguiente tabla 3.1.

Grupo	Endpoint	Descripción
Schedule	`POST /schedule/add_task`	Crea una nueva tarea periódica.
	`GET /schedule/list_tasks_expired`	Lista todas las tareas que ya han expirado.
	`GET /schedule/list_tasks_scheduled`	Lista todas las tareas programadas.
	`GET /schedule/get_task_result/{task_name}`	Dado el nombre de una tarea programada, se obtienen todos los datos extraídos hasta el momento.
	`DELETE /schedule/delete_task/{task_name}`	Elimina la tarea periódica especificada del almacenamiento de Redis.
	`DELETE /schedule/cancel_task/{task_name}`	Detiene las futuras ejecuciones de la tarea periódica (no la elimina de la base de datos).
Task	`GET /task/{task_id}`	Dado el ID de una tarea de Celery, obtiene los metadatos asociados a esa tarea.
	`GET /task/all_tasks_id`	Obtiene todos los IDs de tareas que actualmente están ejecutando procesos de extracción en tiempo real.
	`DELETE /task/cancel_task/{task_id}`	Dado el ID de una tarea Celery en ejecución, detiene y elimina esa tarea.

Tabla 3.1: Endpoints FastAPI

El grupo ***schedule*** hace referencia a las tareas periódicas que se programan en un intervalo de fechas por RedBeat. En cambio, la agrupación de ***task*** hace referencia a las tareas que Celery ejecuta. Es decir, las tareas de Celery que están ejecutando actualmente el proceso de obtención de datos que son creadas por las tareas programadas.

3.3.3. Contenedor de RedBeat

El **contenedor de RedBeat** contiene el servicio de RedBeat. Es necesario para que se puedan enviar a ejecución las tareas programadas. Este contenedor, además de con el **contenedor Backend** , se comunica también con el contenedor de **contenedor de Celery**, para enviar las tareas programadas , y con el **contenedor de Redis**, que cumple un doble papel: almacenar los metadatos de las tareas programadas y actuar como intermediario para el envío de las tareas.

El **componente de RedBeat** permite programar las tareas dado los intervalos de fechas. Este componente actúa como planificador de Celery. Los metadatos (nombre, intervalo de ejecución, periódico digital) de las tareas programadas se almacenan en el llamado **componente de RedisBackend**. Cuando llega el momento de la ejecución de la tarea programada, RedBeat lee del componente de RedisBackend y envía la tarea al broker de Celery, en este caso el **componente RedisBroker**.

3.3.4. Contenedor de Celery

El **contenedor de Celery** corre el servicio de Celery necesario para que las tareas programadas se ejecuten. Además, contiene la lógica necesaria para llevar a cabo el proceso de extracción de datos (web scraping). Contiene tanto la parte de obtención del código de la web, mediante la librería Crawl4ai, como la extracción de datos, mediante el uso del LLM (Large Language Model) a través de la librería LangChain. Este contenedor se comunica con el **contenedor de Redis**, que lo utiliza como broker para distribuir las tareas, con el **contenedor de MongoDB** para guardar los datos extraídos de los procesos de web scraping, y por último con el **contenedor Backend**.

El **Componente de CeleryWorker** encapsula toda la lógica necesaria para ejecutar las tareas programadas y para ejecutar el proceso de extracción de datos con el **componente de Crawl4AI** y el **componente de Langchain**. Los workers leen de **RedisBroker** la tarea programada y la ejecutan en la hora planificada. Para ello, Celery crea sus propias tareas que ejecutan todo este proceso de extracción en tiempo real. Este es el punto donde comienza el proceso de obtener los datos: título, autor, lead, url y date de las noticias de portada del periódico digital.

3.3.4.1. Scraping Adaptativo

Gracias a mezclar las librerías de Crawl4AI y Langchain podemos crear un proceso de extracción de datos adaptativo, el punto clave de esta aplicación. Por una parte, el **componente de Crawl4AI** contiene todas las funciones que nos ayudan a obtener el código fuente de la portada del medio digital y las que nos ayudan a convertir ese código en algo con lo que el LLM puede trabajar. Por otro lado, el **componente de Langchain** contiene toda la lógica necesaria para poder extraer todos los datos de las noticias de portada. A continuación, explicamos el funcionamiento con más detalle apoyándonos en este pseudo código.

```
def scraping_adaptativo(url_portada):

    # 1. PARTE CRAWL4AI: obtener y preparar contenido

    codigo_html = Crawl4AI.obtener_html(url_portada)
    codigo_filtrado = Crawl4AI.eliminar_elementos_inutiles(codigo_html)
    markdown = Crawl4AI.convertir_a_markdown(codigo_filtrado)

    secciones = Langchain.segmentar_por_encabezados(markdown)
    fragmentos = Langchain.dividir_en_chunks(secciones, solapamiento=True)

    # 2. PARTE LANGCHAIN: extraer datos con LLM

    modelo = Langchain.cargar_modelo(nombre="LLaMa3:8B", temperatura=0)
    prompt = Langchain.definir_prompt(prompt=extraer_datos,formato_salida="JSON")


    for fragmento in fragmentos:
        respuesta = Langchain.ejecutar_chain(fragmento, prompt, modelo)
        json_datos = Langchain.parsear_salida(respuesta)
        MongoDB.insertar(json_datos)
```

En el **componente de Crawl4AI**, inicialmente recolectamos todos los elementos de la página web utilizando la biblioteca Crawl4ai, que nos permite obtener tanto los elementos dinámicos como los elementos HTML estáticos. Además, nos permite especificar etiquetas HTML que podemos ignorar, así nos quitamos elementos inútiles. El resultado final de este proceso es un Markdown que contiene toda la portada del periódico digital. Este Markdown se divide en pequeños trocitos que serán enviados al **componente de Langchain**. La razón de esta división es porque los modelos LLM tienen una ventana de contexto, es decir, un límite del tamaño del prompt. Para realizar esta división, utilizamos herramientas de la librería Langchain que nos permiten segmentar el texto. Primero, segmentamos el documento respetando su estructura mediante encabezados Markdown. Luego, dividimos en fragmentos más pequeños con solapamiento entre ellos para mantener la coherencia contextual. Así, tenemos una colección de fragmentos optimizados para la extracción de datos.

En el **componente de Langchain** primero se establece la configuración del LLM. Hay muchas formas de configuración, pero en nuestro caso hemos utilizado Ollama, por lo que el modelo LLM debe estar cargado en el dispositivo, con el modelo LLaMa3:8B de META y una temperatura igual a 0, para que se centre exclusivamente en los datos que le proporcionamos y su comportamiento no sea tan impredecible. Por último, en la configuración, también hemos especificado que el tipo de respuesta del modelo sea en formato JSON, facilitando así la extracción de los datos de las noticias en portada. Después, definimos el prompt para que siga las indicaciones necesarias para obtener los datos de las diferentes noticias de portada. A continuación se explican las partes del prompt:

```
"""
You are an expert web scraper.
I want you to analyze the following text written in Markdown format,
which represents a series of news articles from a digital newspaper.

Your task is to extract the following information for **each news article**:

1. **Title**: The text inside the second-level heading (`##`) which also
    contains a link.
2. **Author**: The name of the author, which appears just below the headline,
    usually as a hyperlink.
3. **Lead paragraph**: The plain text that appears immediately after the author,
    summarizing or introducing the article.
4. **URL**: The hyperlink associated with the headline.
5. **Date**: Extract the date from the article's URL in `YYYY-MM-DD` format
    (e.g., if the URL is `https://elpais.com/espana/2025-04-16/...`,
        the date is `2025-04-16`).

Present the extracted data in the following structured JSON format:
    {{
        "title": "",
        "author": "",
        "lead": "",
        "url": "",
        "date": ""
    }}

DONT MAKE THINGS UP. IF YOU CAN NOT EXTRACT THE DATA LEAVE THE FIELD EMPTY.
JUST RETURN THE JSON DATA WITHOUT ANY ADDITIONAL TEXT OR EXPLANATION.
"""
```

Buscamos darle un contexto al modelo y ser precisos y detallados con la tarea limitando la salida. Primero se le asigna un papel, que en este caso es el de experto en web scraping, lo cual proporciona al modelo el contexto necesario para determinar su comportamiento. Segundo, se le dan instrucciones claras y específicas de la tarea que tiene que llevar a cabo: extraer los datos del Markdown y dónde puede encontrar cada uno. Tercero, se le define un tipo de salida concreto, formato JSON con los campos título, autor, lead, url y date. Esto nos va a facilitar mucho el trabajo de inserción en la base de datos. Por último, le damos instrucciones para evitar alucinaciones y limitamos que la salida sea únicamente el JSON que le hemos pedido.

Después de definir el prompt, se crean las cadenas (chains). Estas cadenas representan el flujo que siguen nuestros trozos de Markdown (chunks) desde que entran hasta obtener la salida final. En este caso, conectamos el prompt con el modelo del lenguaje y la salida obtenida la convertimos en un objeto JSON, con los campos necesarios para poder hacer la inserción en la base de datos. Tener definida una cadena nos permite repetir este proceso, facilitando así la extracción de todos los chunks de nuestro periódico digital.

3.3.5. Contenedor de Redis

El **contenedor de Redis** contiene el servicio de Redis necesario para llevar a cabo toda la lógica del sistema de gestión de tareas. Existe una dualidad con Redis, ya que actúa tanto como broker para distribuir las tareas a los workers de Celery, como almacenamiento de las tareas programadas por RedBeat.

3.3.6. Contenedor de MongoDB

El **contenedor de MongoDB** contiene el servicio de la base de datos NoSQL de MongoDB donde se guardan todos los datos de las noticias extraídos de los procesos de web scraping de las tareas programadas.

El **componente MongoDB** alberga toda la lógica necesaria para almacenar los datos de las noticias obtenidos de los procesos de web scraping. Todos estos datos se almacenarán en una única colección y se añadirá un campo para identificar a qué tarea programada pertenece. El esquema de los documentos en la colección incluye los siguientes campos:

- **_id**: Identificador único del documento (generado automáticamente por MongoDB).
- **Title**: Título de la noticia.
- **Author**: Autor del artículo.
- **Lead**: Entradilla o resumen breve de la noticia.
- **URL**: Enlace a la noticia original.
- **Date**: Fecha de publicación de la noticia.
- **Task_name**: Nombre de la tarea programada por el usuario.

Capítulo 4

Resultados

En este apartado analizaremos los resultados obtenidos mediante la comparación de las portadas, con la información extraída y guardada en la base de datos.

4.1. Interfaz de Usuario

La interfaz de usuario de *NewScrAIper* se caracteriza por su diseño simple, centrándose en la funcionalidad y la facilidad de uso. La figura 4.1 muestra la pantalla de inicio de la aplicación que, a su vez, es la ventana de programar tareas. Además, se puede observar un menú de navegación con las siguientes secciones:

- **Create Task**: Pantalla inicial y ventana para configurar nuevas tareas de web scraping.
- **List Expired Tasks**: Muestra las tareas programadas que han finalizado.
- **List Scheduled Tasks**: Muestra las tareas programadas que aún no han finalizado.
- **Get Task Data**: Permite recuperar y visualizar los datos extraídos por una tarea específica.

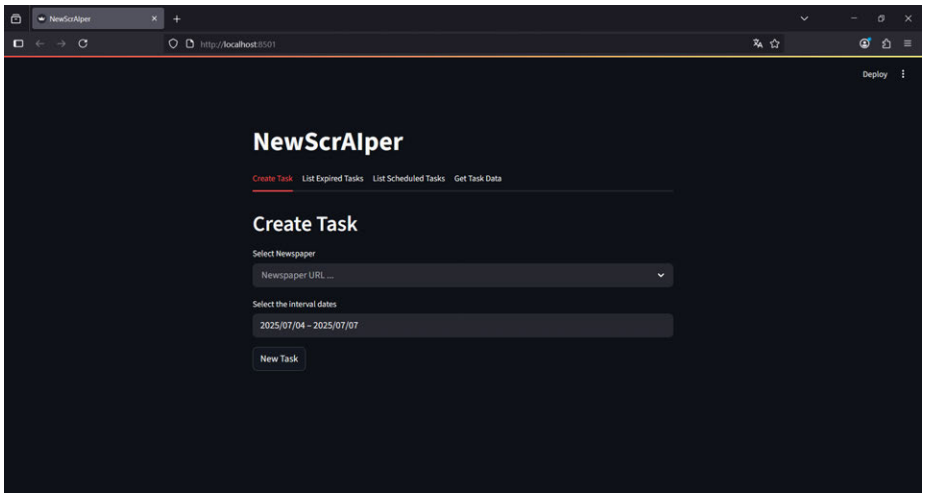


Figura 4.1: *Interfaz Pantalla Principal*

4.1.1. Create Task

En esta ventana, figura 4.1, el usuario puede definir una nueva tarea de web scraping seleccionando el periódico digital y el rango de fechas. Los componentes de esta ventana son:

- **Select Newspaper** : Menú desplegable que permite al usuario elegir el medio digital. Al hacer clic en el campo, se despliega una lista de URLs predefinidas como se puede ver en la figura 4.2).

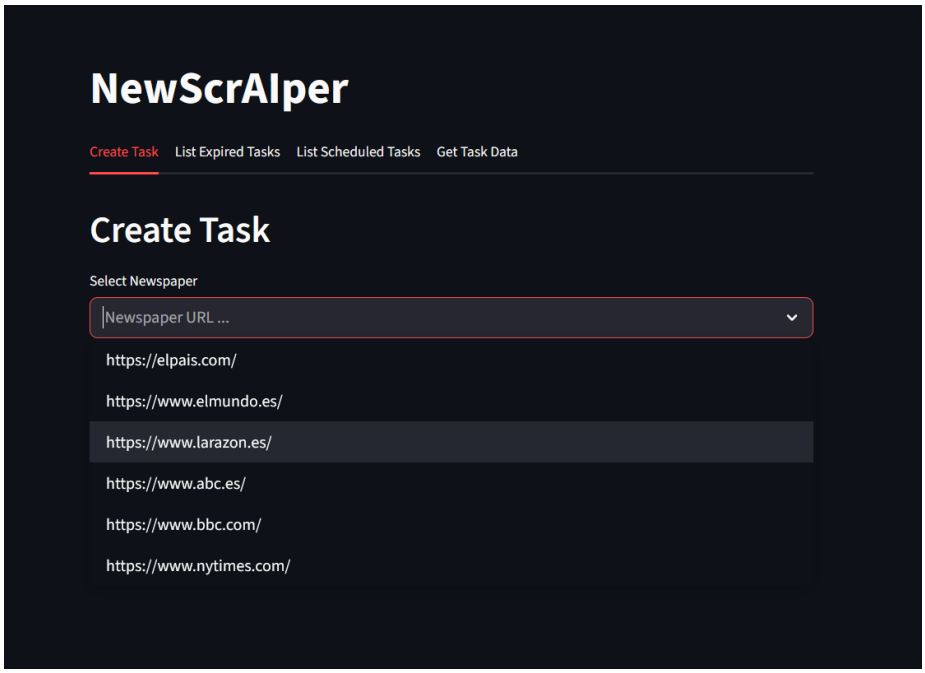


Figura 4.2: *Interfaz Seleccionar Periódico*

- **Select the interval dates**: Campo que permite definir el rango de fechas. Al hacer clic, se despliega un calendario interactivo, ver figura 4.3 donde puedes seleccionar cualquier fecha a partir del día actual.

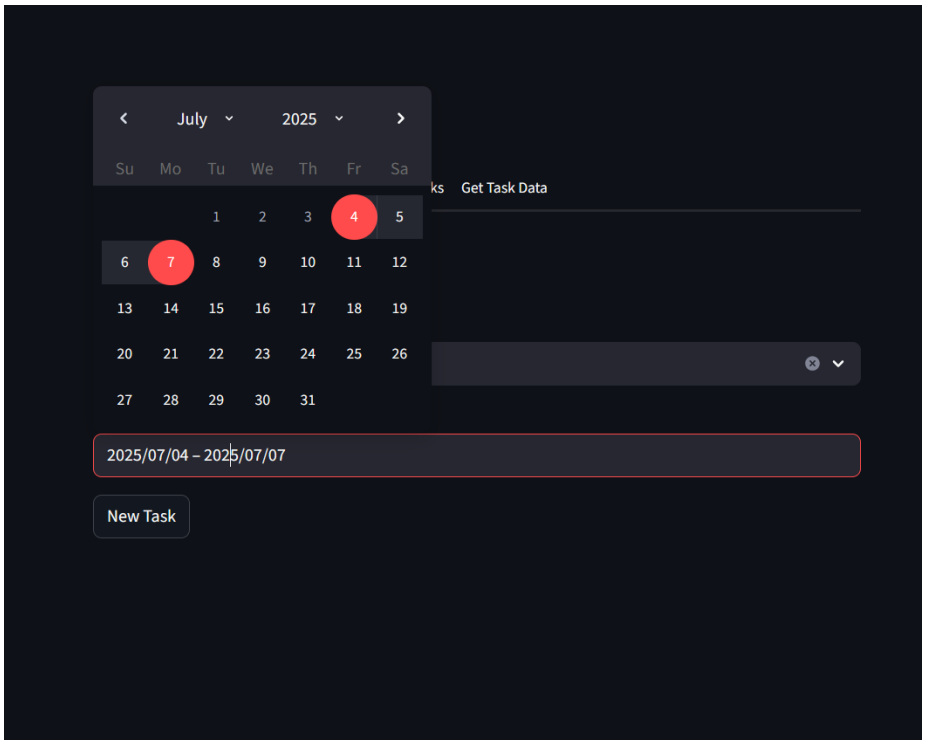


Figura 4.3: *Interfaz Seleccionar Fechas*

- **New Task**: Una vez seleccionados el periódico y las fechas, al hacer clic en este botón se crea y programa la tarea. Como se muestra en la figura 4.4, se presenta un mensaje de éxito indicando el nombre, horario (12:00h) y rango de fechas. El nombre de la tarea, `''task_name''`, se genera automaticamente con el siguiente formato : `crawl-{url-periodico}--{fecha-inicio}-to-{fecha-fin}`.

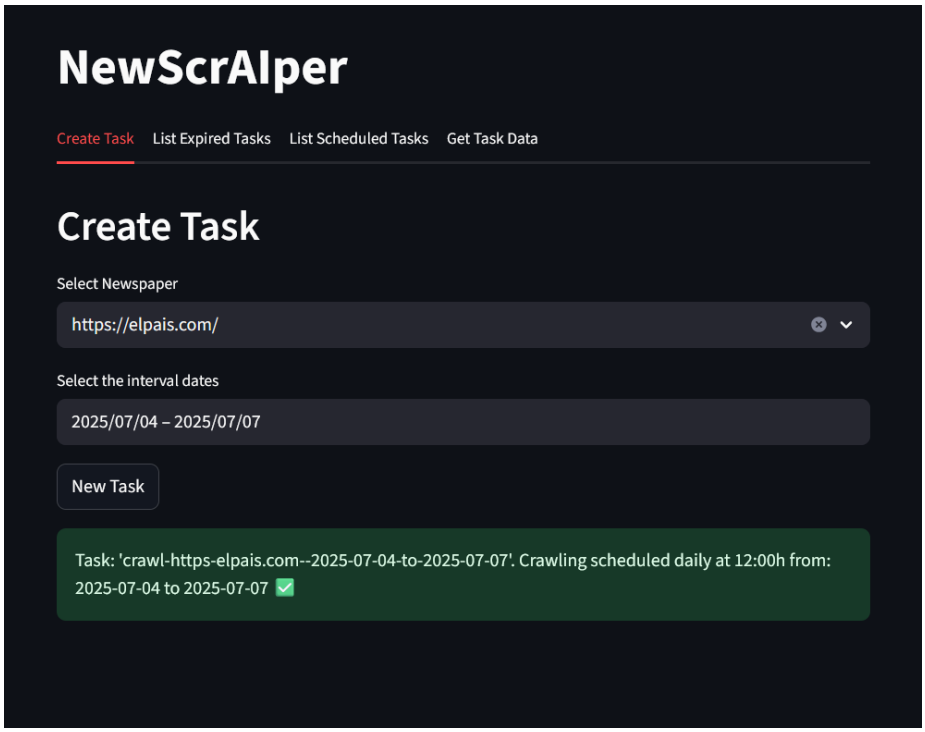


Figura 4.4: *Interfaz Tarea Programada*

Para este MVP se ha decidido que el proceso de scraping solamente se ejecute una vez al día a las 12:00 h, horario de verano español.

4.1.2. List Expired Tasks

En esta sección, el usuario puede consultar las tareas de scraping que ya han expirado. Sus elementos son:

- **List Expired Tasks**: Botón que muestra la lista de tareas expiradas en formato JSON (ver figura 4.5). Cada entrada incluye los campos: ‘‘`task_name`’’, con el nombre de la tarea, y ‘‘`args`’’, con el enlace del periódico digital.

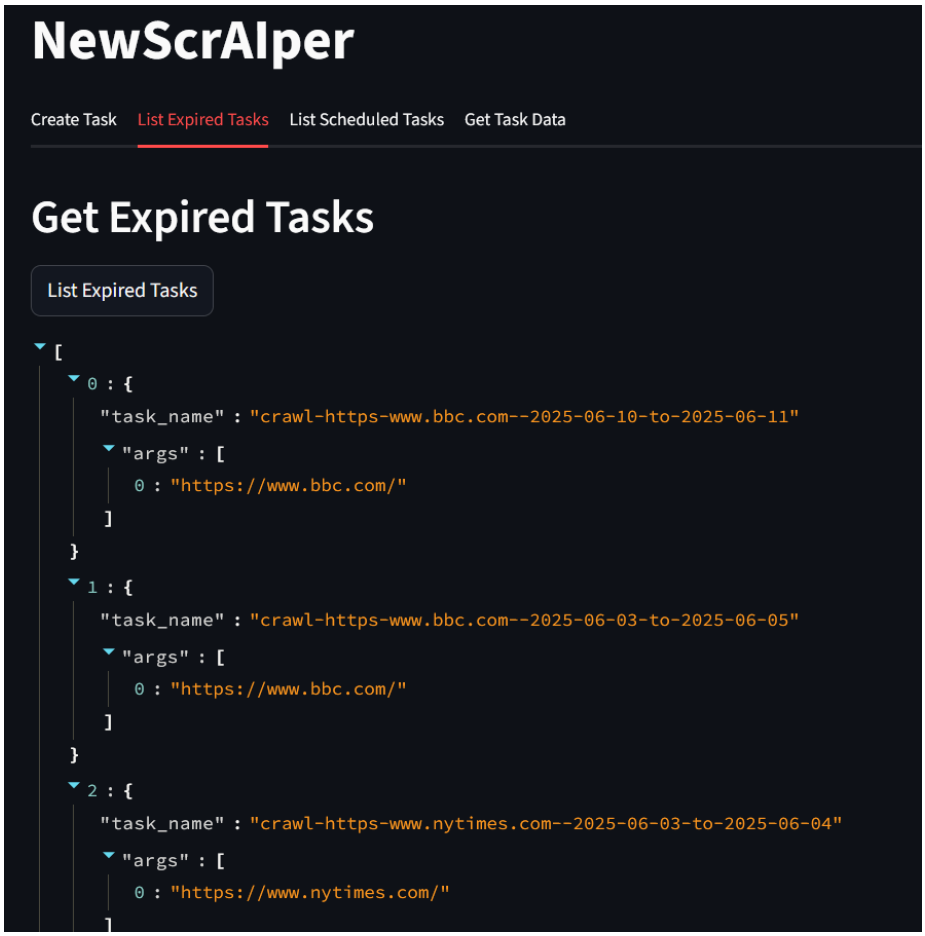


Figura 4.5: *Interfaz Tareas Expiradas*

4.1.3. List Scheduled Tasks

Permite ver y gestionar las tareas programadas activas. Tiene los siguientes elementos:

- **List Schedule Tasks**: Botón que muestra la lista en formato JSON (ver figura 4.6). Incluye los campos: ‘‘`task_name`’’, ‘‘`options`’’ (fecha de expiración), ‘‘`schedule`’’ (horario), y ‘‘`last_run_at`’’ (última ejecución).

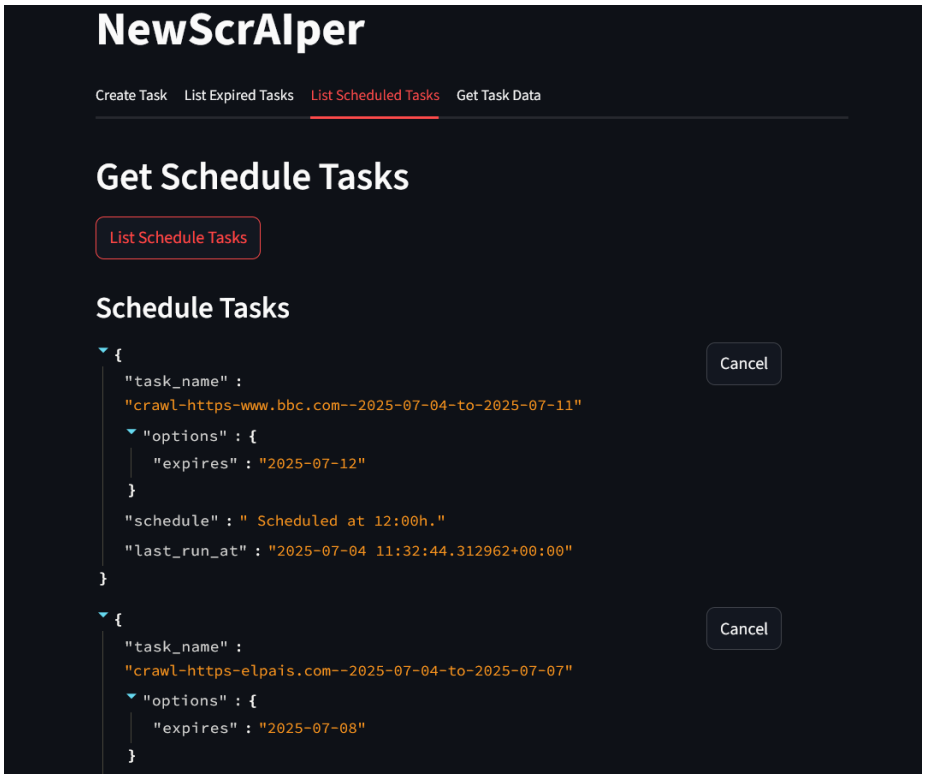


Figura 4.6: *Interfaz Lista Tareas Programadas*

- **Cancel**: Botón junto a cada tarea para cancelarla. Al hacerlo, aparece un mensaje de confirmación como muestra la figura 4.7.

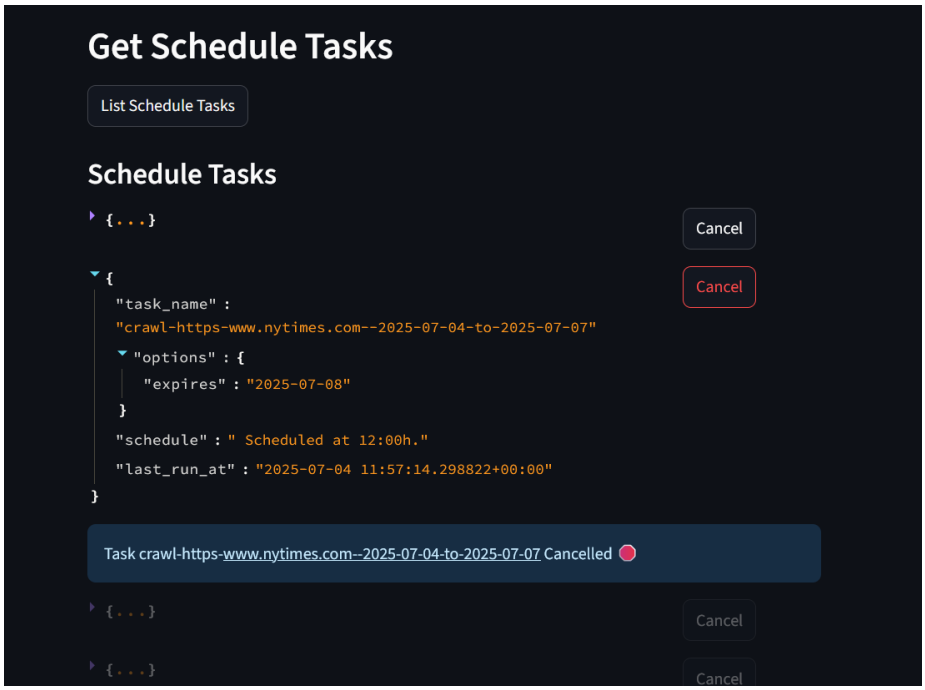


Figura 4.7: Interfaz Tarea Cancelada

4.1.4. Get Task Data

Sección para acceder a los datos extraídos por tareas específicas. Los elementos de esta sección son:

- **Task Name**: Campo para introducir el ‘‘`task_name`’’, nombre de la tarea programada (ver figura 4.8).
  Ejemplo: `crawl-https-elpais.com--2025-06-03-to-2025-06-05` .

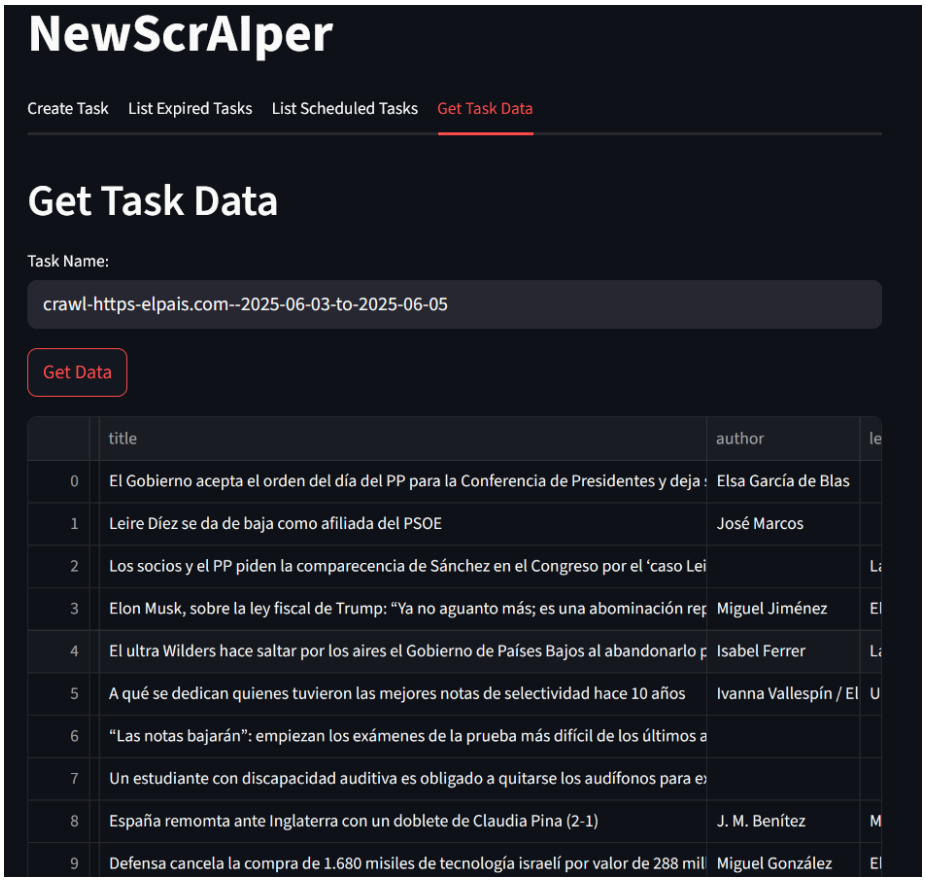


Figura 4.8: Interfaz Obtener Datos

- **Get Data**: Botón que recupera los datos y los presenta en una tabla. Las columnas incluyen ‘‘`title`’’, ‘‘`author`’’, ‘‘`lead`’’ y ‘‘`url`’’ (ver figuras 4.8 y 4.9).
- **Controles de la Tabla** (ver figura 4.10):
  En la propia tabla podemos ejecutar diferentes funcionalidades:
  - **Sort ascending/descending**: Reorganiza los datos de una columna manera alfabética de manera ascendente o descendente.
  - **Pin column**: Fija la columna a un lado.
  - **Hide column**: Oculta la columna.
  - **Autosize (Ajustar automáticamente)**: Ajusta el ancho al contenido.

  Además, en la parte superior derecha de la tabla vemos un conjunto de íconos con las siguientes funcionalidades ordenadas de izquierda a derecha:
  - **Descargar**
  - **Buscar en la tabla**
  - **Ampliar el tamaño de la tabla**

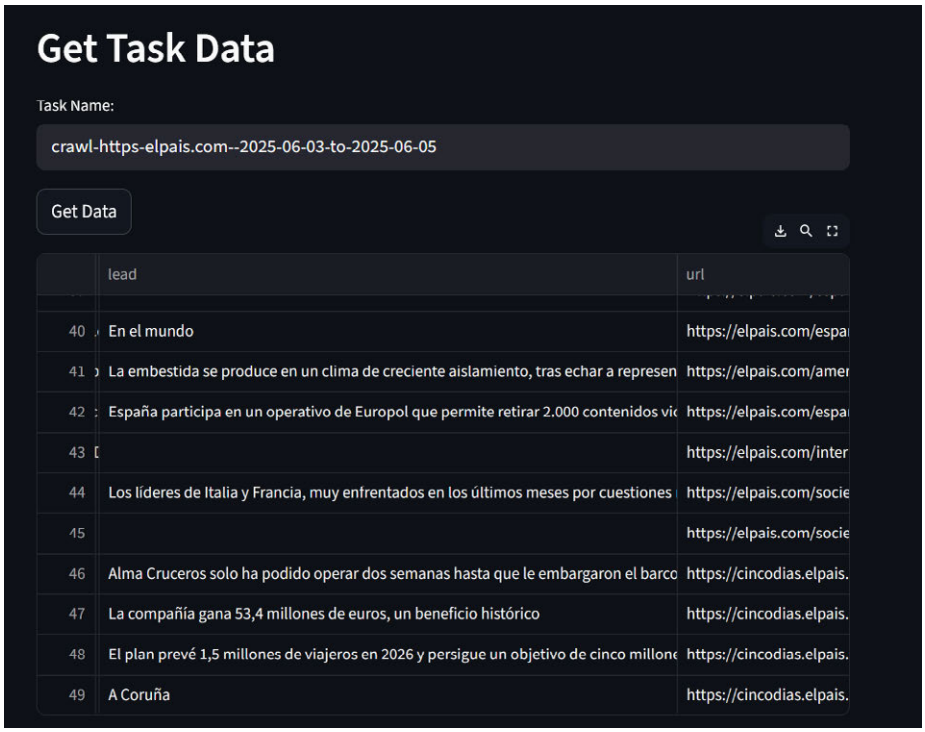


Figura 4.9: Interfaz Obtener Datos

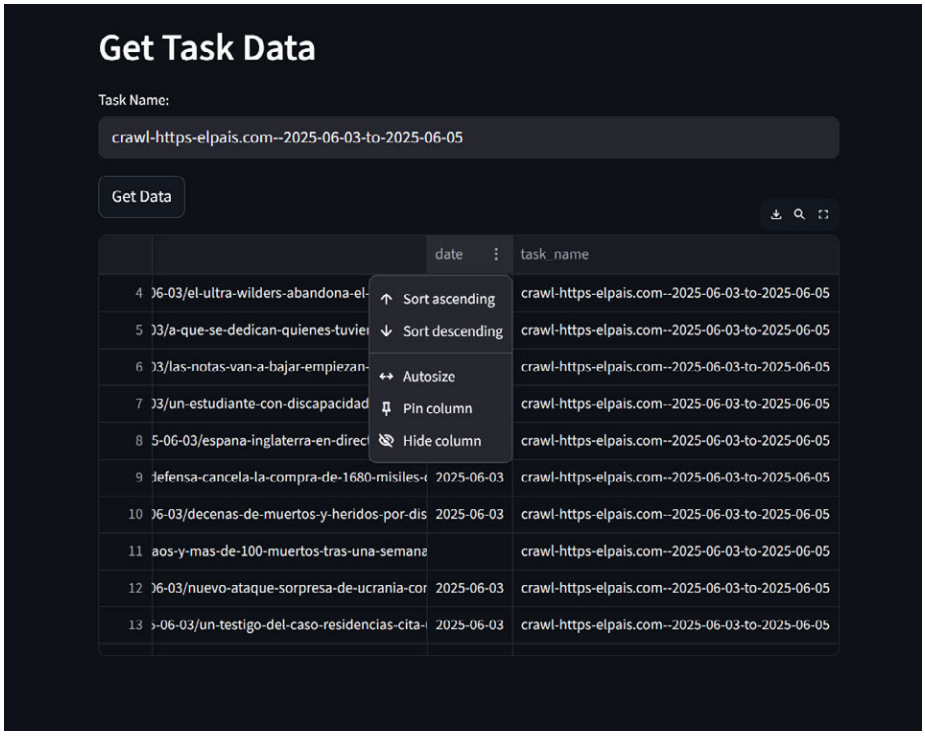


Figura 4.10: Interfaz Obtener Datos Toolkit Tabla

4.2. Casos de Uso

Para describir las interacciones del usuario con el sistema, nos apoyaremos en el diagrama de casos de uso de la figura 4.11.

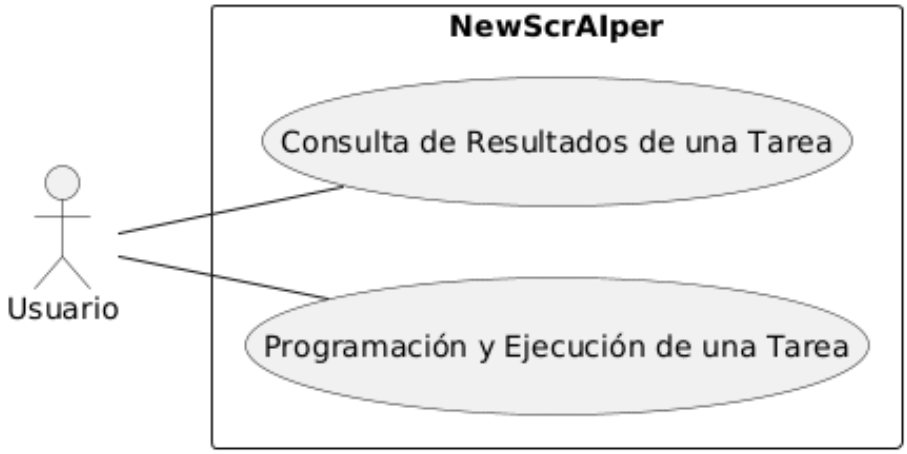


Figura 4.11: Diagrama Casos de Uso

A continuación, se presenta una descripción a alto nivel de los dos casos de uso: Programación y Ejecución de una Tarea y Consulta de Resultados de una Tarea:

1. Caso de Uso: Programación y Ejecución de una Tarea

- **Nombre:** Programación y Ejecución de una Tarea,
- **Actores:** Usuario
- **Tipo:** Primario
- **Descripción:** Este caso de uso permite al usuario configurar una nueva tarea de web scraping. El usuario selecciona el medio digital y el rango de fechas para la extracción de datos. Una vez definida, la tarea es programada para su ejecución periódica en el rango de fechas. Al llegar la hora programada, el sistema automáticamente inicia el proceso de scraping, extrayendo los datos de titular, autor, lead, fecha y URL de las noticias del medio seleccionado y almacenándola en la base de datos para su posterior consulta.

Para entender mejor el funcionamiento del sistema en este caso, se han creado los siguientes diagramas secuenciales:

- **Diagrama Secuencial Programar Tarea**: Representa la interacciones de los componentes para crear una tarea. Figura 4.12

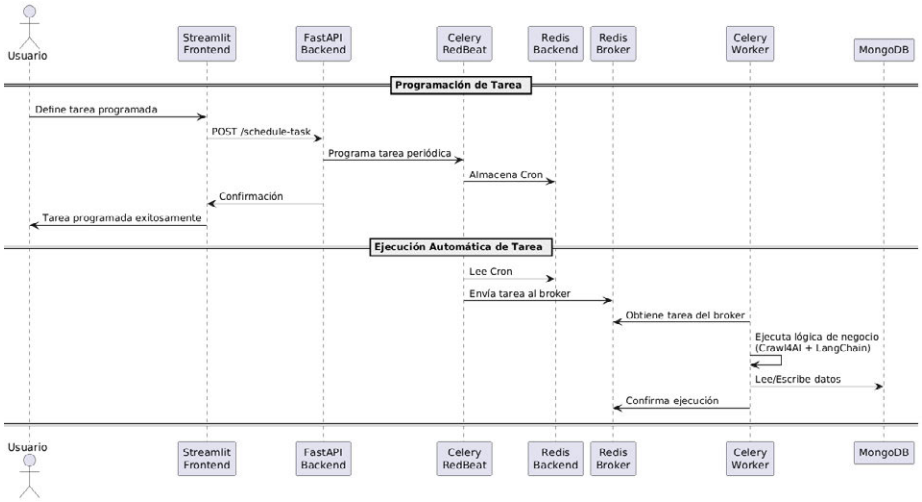


Figura 4.12: *Diagrama Secuencial Programar Tarea*

- **Diagram Secuencial Scraping**: Representa la interacciones de los componentes involucrados en el proceso de extración de datos de las noticias de portada. Figura 4.13.

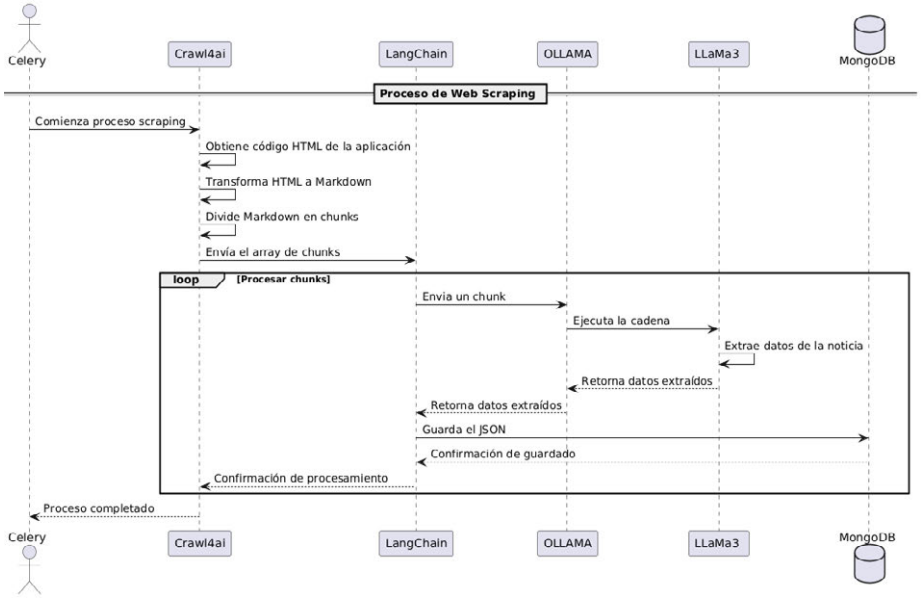


Figura 4.13: Diagrama Secuencial Extracción de Datos

2. Caso de Uso: Consulta de Resultados de una Tarea

- **Nombre:** Consultar Resultados de Tarea
- **Actores:** Usuario

- **Tipo:** Primario
- **Descripción:** Este caso de uso permite al usuario obtener y visualizar los datos que han sido extraídos por una tarea previamente ejecutada. El usuario proporciona el nombre de la tarea deseada, y el sistema muestra los resultados en una tabla interactiva, permitiendo ordenar, buscar o descargar la información extraida.

4.3. Resultados obtenidos

En esta sección nos centraremos en el caso de uso **2. Consulta de Resultados de una Tarea**. Analizaremos los datos obtenidos (titular, autor, lead, fecha y URL) de las noticias de portada de los periódicos: La Razón, El País, El Mundo y ABC. Dado que revisar la totalidad de las noticias resultaría muy extenso debido a la gran cantidad de noticias en las portadas, nos enfocaremos en los titulares principales y resaltaremos los errores más significativos. Una noticia será considerada correctamente procesada si todos los campos esperados (titular, autor, lead, fecha de publicación y URL de la noticia) son obtenidos de manera precisa y fiel a su representación original en la portada del periódico, sin omisiones, errores semánticos o adiciones no deseadas que alteren el contenido de los datos.

4.3.1. El País

A continuación se muestra la portada del periódico El País de la fecha 2025/07/07 enumerando y destacando las noticias donde no se han obtenido correctamente los datos.

Figura 4.14: El País 07/07/2025

Figura 4.15: El País 07/07/2025

Figura 4.16: El País 07/07/2025

EL PAÍS

SUSCRÍBETE

JESÚS JANK CURBELO | Kerrville

Las riadas en el condado de Kerr han dejado al menos 80 fallecidos, entre ellos 28 menores, y decenas de desaparecidos

Así fue la crecida del río Llano en Texas

Roman Starovoit, el 7 de mayo en Moscú. MAXIM SHEMETOV (REUTERS)

Hallado muerto el ministro de Transporte ruso horas después de ser destituido por Putin

JAVIER G. CUESTA | Moscú

Las investigaciones oficiales apuntan a que Starovoit se ha suicidado. Fuentes políticas dicen que el exgobernador de la provincia fronteriza de Kursk estaba a punto de ser acusado de corrupción

Más noticias

Un incendio forestal "de alta intensidad" obliga a confinar cuatro municipios de Tarragona

Detenido un pederasta convicto por agredir sexualmente y ofrecer regalos y juegos a menores durante sus permisos

Ayuso, sobre sus universidades públicas: "Vamos a reforzar significativamente la inversión a seis años"

Detenido en Espartinas (Sevilla) el maltratador que se cambió de género tras una condena

LA SALUD DEL FUTURO UN PROYECTO DE Sanitas

Una resonancia más amable, en menos tiempo y sin sensación de claustrofobia. Tecnología de vanguardia al servicio del paciente

Del 3 al 16 de julio de 2025
El verano de las marcas en Hipercor
10€
DE REGALO*
por cada 50€ de compra en más de 250 marcas
*Canjea un cupón de 10€ por cada 60€ de compra en alimentación, droguería y perfumería en Hipercor, del 17 de julio al 13 de agosto de 2025.
COMPRAR
HIPERCOR

4.3.1.1. Primer grupo (figura 4.14)

Empezaremos analizando el grupo de noticias: **1.‘‘La Audiencia imputa a Pardo de Vera y Javier Herrero por el supuesto amaño de obra pública’’, 2. ‘‘El dueño de la empresa navarra respalda la versión de Santos Cerdán y niega el pago de mordidas’’ 3.‘‘Los audios del Comité Federal del PSOE a puerta cerrada: Es un ejercicio de hipocresía insoportable ’’**. A continuación, se muestran los documentos de la base de datos con los datos extraídos respecto a este grupo:

- **1º Documento**
  Título: *La Audiencia imputa a Pardo de Vera y Javier Herrero por el supuesto amaño de obra pública*
  Autor: J. J. Gálvez
  Lead: *vacio*
  Fecha: 2025-07-07
  Enlace:https://elpais.com/espana/2025-07-07/la-audiencia-nacional-imputa-a-isabel-pardo-de-vera-y-javier-herrero-por-el-supuesto-amano-de-obra-publica-del-caso-cerdan.html

- **2º Documento**
  Título: *El dueño de la empresa navarra respalda la versión de Santos Cerdán y niega el pago de mordidas*
  Autor: J. J. Gálvez
  Lead: *vacio*
  Fecha: 2025-07-07
  Enlace:https://elpais.com/espana/2025-07-07/el-dueno-de-la-empresa

-navarra-respalda-la-version-de-santos
-cerdan-y-niega-el-pago-de-mordidas.html

- **3º Documento**
  Título: *Un exdirectivo de Acciona también rechaza los sobornos. El Supremo les prohíbe salir del país*
  Autor: José Marcos
  Lead: *vacio*
  Fecha: 2025-07-07
  Enlace:https://elpais.com/espana/2025-07-07/los-audios-del-comite-federal
  -del-psoe-page-aguantar-no-deja-de-ser-cavar
  -mas-un-agujero-puente-le-pido-no-le-exijo-al
  -secretario-general-que-siga.html

- **4º Documento**
  Título: *El presidente de Castilla-La Mancha, García-Page, recibió la réplica a sus argumentos del ministro de Transportes, muy aplaudido en sus intervenciones*
  Autor: *vacio*
  Lead: El presidente de Castilla-La Mancha, García-Page, recibió la réplica a sus argumentos del ministro de Transportes, muy aplaudido en sus intervenciones
  Fecha: 2025-07-07
  Enlace:https://elpais.com/espana/2025-07-07/los-audios
  -del-comite-federal-del-psoe-ines-rey-ojo-con
  -los-acantilados-de-cristal-las-mismas
  -estructuras-patriarcales-a-veces-las-tenemos-aqui.html

Como podemos observar, los datos guardados de los documentos no son fieles a lo que se muestra en la portada.

Análisis de Documentos

- **Documentos 1 y 2:** Corresponden a la noticia 1 y noticia 2, respectivamente. En ambos casos, el **lead** no se ha extraído correctamente.
- **Documento 3:** La **URL** de este documento es la de la noticia 3, mientras que el **título** de este documento se corresponde con el lead de la noticia 2.
- **Documento 4:** La **URL** de este documento es de una noticia diferente (”Las socialistas alzaron la voz: “Hablaban de las mujeres de manera asquerosa””). Pero el **título** y el **lead** de este documento corresponden al lead de la noticia 3.

Motivo

El problema principal radica en la **estructura desordenada de las separaciones de Markdown (chunks)**. A esto se suma la **baja capacidad**

del modelo utilizado (8B de parámetros), lo que limita al momento de procesar y organizar la información de manera óptima en comparación con modelos de mayor tamaño.

Explicación

Para esta explicación, nos apoyaremos en la representación de los chunks que se muestran a continuación.

- El **lead de la noticia 1** se encuentra en el **chunk 5**, mientras que el resto de los datos de esa noticia están en el **chunk 4**.
- De manera similar, los datos de la **noticia 2** están en el **chunk 5**, pero su **lead** se ubica en el **chunk 6**.
- El mismo patrón se repite con la **noticia 3**: el **chunk 6** contiene sus datos, pero el **lead** está en el **chunk 7**, el cual, a su vez, contiene información de otra noticia.
- A partir del **chunk 8**, la estructura de los datos se normaliza y presenta un orden más coherente.

El modelo intenta alinear los datos de las noticias lo más fielmente posible a la prompt dada, pero la división de los chunks impide que la extracción sea completamente satisfactoria.

Chunk 4

```
----Chunk-4----
[07 jul 2025|Actualizado 15:21 CEST|[Exprés](https://elpais.com/
expres/?rel=cab) ## [La Audiencia imputa a Pardo de Vera
y Javier Herrero por el supuesto amaño de obra pública]
(https://elpais.com/espana/2025-07-07/la-audiencia-nacional-imputa-
a-isabel-pardo-de-vera-y-javier-herrero-por-el-supuesto-amano-de-obra-
publica-del-caso-cerdan.html) [J. J. Gálvez](https://elpais.com/autor/
jose-maria-jimenez-galvez/) | Madrid],
```

Chunk 5

```
---Chunk-5----
[El magistrado cita a la expresidenta de Adif y al ex director general
de Carreteras a declarar el 21 de julio ## [El dueño de la empresa
navarra respalda la versión de Santos Cerdán y niega el pago de mordidas]
(https://elpais.com/espana/2025-07-07/el-dueno-de-la-empresa-navarra-
respalda-la-version-de-santos-cerdan-y-
niega-el-pago-de-mordidas.html) [J. J. Gálvez](https://elpais.com/autor/
jose-maria-jimenez-galvez/) | Madrid],
```

Chunk 6

```
---Chunk-6----
[Un exdirectivo de Acciona también rechaza los sobornos. El Supremo
les prohíbe salir del país ## [Los audios del Comité Federal del
PSOE a puerta cerrada: \Es un ejercicio de hipocresía insoportable"]
(https://elpais.com/espana/2025-07-07/
los-audios-del-comite-federal-del-psoe-page-aguantar-no-deja-de-ser-cavar-
mas-un-agujero-puente-le-pido-no-le-exijo-al-secretario-general-
que-siga.html) [José Marcos](https://elpais.com/autor/jose-marcos/) |
Madrid],
```

Chunk 7

```
---Chunk-7----
[El presidente de Castilla-La Mancha, García-Page, recibió la réplica a sus
argumentos del ministro de Transportes, muy aplaudido en sus intervenciones
* [Las socialistas alzaron la voz: \Hablaban de las mujeres de manera
asquerosa"](https://elpais.com/espana/2025-07-07/los-audios-del-comite-
federal-del-psoe-ines-rey-ojo-con-los-acantilados-de-cristal-las-
mismas-estructuras-patriarcales-a-veces-las-tenemos-aqui.html)],
```

Chunk 8

```
---Chunk-8----
## [Page responde a Puente: \Yo no tengo puto amo. La mayoría del partido
no piensa como el ministro"] (https://elpais.com/espana/2025-07-07/
page-responde-a-puente-yo-no-tengo-puto-amo-la-inmensa-mayoria-del-
partido-no-piensa-como-el.html) [Manuel Viejo](https://elpais.com/autor/
manuel-viejo/) | Madrid El ministro de Transporte, tras escuchar al
presidente de Castilla-La Mancha: \Yo hablo donde tengo que hacerlo y
mirando a la gente a la cara"],
```

4.3.1.2. Segundo grupo (fig 4.15)

En este caso nos analizaremos la noticia: **‘‘4. Trump y Von der Leyen hablan por teléfono para avanzar hacia un pacto que acabe con la guerra comercial’’**. Los datos que se han extraído son:

- **5º Documento**
  Título: *Trump y Von der Leyen hablan por teléfono para avanzar hacia un pacto que acabe con la guerra comercial*
  Autor: *vacio*
  Lead: *vacio*
  Fecha: *vacio*
  Enlace:https://elpais.com/internacional/2025-07-07/trump-y-von-der-leyen-hablan-por-telefono-para-avanzar-hacia-un-principio-de-acuerdo-que-acabe-con-la-guerra-comercial.html

- **6º Documento**
  Título: *La negociación sigue a 48 horas de que acabe el plazo para alcanzar*

un acuerdo.
Autor: Silvia Ayuso
Lead: *La negociación sigue a 48 horas de que acabe el plazo para alcanzar un acuerdo. EE UU prevé que los aranceles entren en vigor el 1 de agosto*
Fecha: *vacio*
Enlace:`https://elpais.com/bruselas/`

Como podemos observar, los datos almacenados de los documentos no coinciden exactamente con lo que se muestra en la portada.

Análisis de Documentos

- **Documento 5:** El título y el enlace están bien extraídos, pero el autor, el lead y la fecha están vacíos.
- **Documento 6:** El título pertenece al lead de la noticia 4 y el autor pertenece al autor de la noticia 4. El lead de este documento pertenece al lead de la noticia 4. La fecha también está vacía y la URL es incorrecta.

Motivo

La problemática se repite. La estructura de los chunks está desordenada y la capacidad del LLM de ordenar y extraer los datos correctamente en estos casos es mala. Por eso, los datos obtenidos no aparecen donde deben y como deben. Además, el modelo no reconoce el patrón de la fecha en la URL.

Explicación

Fijándonos en la representación de los chunks referentes a esta noticia, podemos observar cómo en el **chunk 15** se encuentra el lead de la **noticia 4**, mientras que el resto de los datos se encuentra en el **chunk 14**. El modelo intenta encajar

Chunk-14

```
----Chunk-14---
[07 jul 2025|Actualizado 15:21 CEST|[Exprés](https://elpais.com/
expres/?rel=cab) ## [Trump y Von der Leyen hablan por teléfono
para avanzar hacia un pacto que acabe con la guerra comercial]
(https://elpais.com/internacional/2025-07-07/
trump-y-von-der-leyen-hablan-por-telefono-para-avanzar-hacia-un-
principio-de-acuerdo-que-acabe-con-la-guerra-comercial.html) ],
```

Chunk-15

```
---Chunk-15---
[La negociación sigue a 48 horas de que acabe el plazo para alcanzar un
acuerdo. Silvia Ayuso La negociación sigue a 48 horas de que acabe el
plazo para alcanzar un acuerdo. EE UU prevé que los aranceles entren en
vigor el 1 de agosto (https://elpais.com/bruselas/)],
```

4.3.1.3. Tercer grupo (fig 4.16)

Analizaremos ahora el grupo de noticias: **5.‘‘Una sentencia pionera cierra 10 pisos turísticos de un edificio para proteger la salud y la intimidad de los niños de una familia’’, 6. ‘‘Aplazado el desahucio del escritor y periodista Donato Ndongo en Murcia hasta el 23 de julio’’**. A continuación, se muestran los documentos de la base de datos con los datos extraídos respecto a este grupo:

- **7º Documento**
  Título: *Una sentencia pionera cierra 10 pisos turísticos de un edificio para proteger la salud y la intimidad de los niños de una familia*
  Autor: Braulio García Jaén
  Lead: *vacio*
  Fecha: 2025-07-07
  Enlace:https://elpais.com/espana/madrid/2025-07-07/la-justicia-cierra-diez-pisos-turisticos-de-un-edificio-en-madrid-para-proteger-la-salud-y-la-intimidad-de-una-familia-con-dos-ninos.html

- **8º Documento**
  Título: *Aplazado el desahucio del escritor y periodista Donato Ndongo en Murcia hasta el 23 de julio*
  Autor: Virginia Vadillo
  Lead: *vacio*
  Fecha: 2025-07-07
  Enlace:https://elpais.com/espana/2025-07-07/paralizan-el-desahucio-del-escritor-y-periodista-donato-ndongo-hasta-el-23-de-julio.html

Se puede ver que la información almacenada de los documentos no coincide con lo que se presenta en la portada.

Análisis de Documentos

- **Documento 7 y 8:** En este caso, el lead de ambos documentos está vacío.

Motivo

Aquí vuelve a surgir el mismo problema. A pesar de que los elementos están presentes, el modelo no extrae correctamente todos los datos debido a la estructura desordenada de los chunks.

Explicación

Para esta explicación, nos apoyaremos en la representación de los chunks que se muestran a continuación.

- El **chunk 23** contiene la **noticia 5** con su título, enlace y autor, pero no incluye el lead.
- El **chunk 24** contiene el **lead** de la **noticia 5**. Además, contiene la **noticia 6** con su título, enlace y autor, pero nuevamente, no incluye el lead.

- El **chunk 25** contiene el **lead** de la **noticia 6** y además contiene los datos de una noticia difernte.

Chunk 23

```
---Chunk-23----
[## [Una sentencia pionera cierra 10 pisos turísticos de un edificio
para proteger la salud y la intimidad de los niños de una familia]
(https://elpais.com/espana/madrid/2025-07-07/la-justicia-cierra-diez-
pisos-turisticos-de-un-edificio-en-madrid-para-proteger-
la-salud-y-la-intimidad-de-una-familia-con-dos-ninos.html)      [Braulio
García Jaén](https://elpais.com/autor/braulio-garcia-jaen/)|Madrid],
```

Chunk-24

```
---Chunk-24----
[Cuatro empresas deben indemnizar con 37.000 euros a una pareja y sus
dos hijos por el \ruido" y la \insalubridad" causados ## [Aplazado el
desahucio del escritor y periodista Donato Ndongo en Murcia hasta el 23
de julio](https://elpais.com/espana/2025-07-07/paralizan-el-
desahucio-del-escritor-y-periodista-donato-ndongo-
hasta-el-23-de-julio.html) [Virginia Vadillo](https://elpais.com/autor/
virginia-vadillo-jimenez/)|Murcia],
```

Chunk-25

```
---Chunk-25----
[El ecuatoguineano recibe una pensión mínima y reside en la
vivienda desde hace 30 años ## [Una multitud de corredores,
caídas, cinco contusionados y un herido en el primer encierro de
San Fermín](https://elpais.com/cultura/2025-07-07/primer-encierro-de-
san-fermin.html) [Antonio Lorca](https://elpais.com/autor/antonio-lorca/)
La carrera con los veloces toros de Fuente Ymbro ha durado 2 minutos y
37 segundos, 19 más que la de la misma ganadería el año pasado],
```

4.3.2. La Razón

En este apartado analizaremos la portada del 2025/07/04 del periódico La Razón destacando las noticias donde los datos no se han extraído correctamente.

Figura 4.17: La Razón 04/07/2025

Figura 4.18: La Razón 04/07/2025

4.3.2.1. Primer grupo (figura 4.17)

Empezaremos analizando el grupo de noticias: **`1.‘‘Feijóo completa su cúpula con el fichaje de Alberto Nadal’’`**. A continuación, se muestran los documentos de la base de datos con los datos extraídos que tienen relación con esta noticia:

- **1º Documento**
  Título: *Feijóo completa su cúpula con el fichaje de Alberto Nadal*
  Autor: *vacio*
  Lead: *vacio*
  Fecha: *vacio*
  Enlace:`https://www.larazon.es/politica/feijoo-completa-cupula-fichaje-alberto-nadal_202507046867c65977c9307258679640.html`

Análisis de Documentos

- **Documento 1:** Corresponden a las noticia 1.En este caso, el **autor**, **lead** y **fecha** no han sido extraidos.

Motivo

Aquí se identifican tres razones por las que los datos no se están extrayendo correctamente:

- **Extracción del Autor:** El autor no se extrae correctamente porque todas las noticias de la sección resaltada del periódico comparten el mismo autor. Por lo tanto, el nombre del autor solo aparece en el último chunk, no con cada artículo individual.
- **Extracción del Lead:** El lead no se extrae porque este campo se encuentra en un chunk y resto de datos de esta noticia está en otro.
- **Extracción de la Fecha:** La fecha no se extrae porque el modelo no reconoce el patrón de la fecha en la URL.

Explicación

Para esta explicación, nos apoyaremos en la representación de los chunks que se muestran a continuación.

- El **chunk 31** contiene la noticia 1 con su título y enlace, pero no incluye el autor, lead ni fecha.
- El **chunk 32** contiene el **lead** de la noticia 1 y los datos de otra noticia ("Los dos nuevos generales de Feijóo: el único puente con Vox").
- El **chunk 33** contiene el **autor** de todas la noticias de esta sección y los datos de otra noticia ("Una arquitectura para el cambio")

Chunk 31

```
----31----
[Directo: Última hora del XXI Congreso Nacional del PP
https://www.larazon.es/espana/xxi-congreso-nacional-directo-hoy-ultima-
hora-feijoo-ayuso-reacciones_202507046867729d29f2a108e97dc343.html
Congreso Nacional del PP
## [Feijóo completa su cúpula con el fichaje de Alberto Nadal]
https://www.larazon.es/politica/feijoo-completa-cupula-fichaje-alberto-
nadal_202507046867c65977c9307258679640.html
```

Chunk 32

```
----32---- [El líder del PP completa su nueva cúpula, en la
que siguen sus puestos Bendodo, Sémper, Noelia Núñez y la
portavoz en el Senado, Alicia García. Congreso Nacional del PP
### [Los dos nuevos generales de Feijóo: el único puente con
Vox](https://www.larazon.es/politica/dos-nuevos-generales-feijoo-unico
-puente-vox_202507046867a3fa5e724b62f068ce09.html "Los dos nuevos
generales de Feijóo: el único puente con Vox") Ante el congreso del PP],
```

Chunk 33

```
----33---- [**Ante el congreso del PP** ### [Una arquitectura para el
cambio](https://www.larazon.es/espana/arquitectura-cambio_20250704
68671dc677c93072586714a5.html Üna arquitectura para el cambio")
[Pepe Luis Vázquez](https://www.larazon.es/autores/pepe-luis-vazquez/)
[](https://www.larazon.es/espana/page-augura-que-comite-psoe-algunos-
saben-seguiran-cargo-van-tener-que-verse-carcel_202507046867b9
fb29f2a108e97e0e71.html) Remodelación del PSOE],
```

4.3.2.2. Segundo grupo (ver fig 4.18)

En este apartado nos fijaremos en la noticia ***2. "Estas son las obras que la nueva secretaria de Organización del PSOE adjudicó a una de las empresas de la trama Santos Cerdán"***. A continuación se presentan los documentos relacionados con este caso.

- **2º Documento**
  Título: *Las tribulaciones de Sánchez ante su sudoku más endiablado*
  Autor: José Ignacio Wert
  Lead: *vacio*
  Fecha: 2025-07-04
  Enlace:https://www.larazon.es/espana/tribulaciones-sanchez-sudoku-mas-endiablado_202507046866c7655e724b62f0684815.html

- **3º Documento**
  Título: *Rebeca Torró, nueva secretaria de la Organización del PSOE y adjudica más de 35 millones a empresas*
  Autor: Christian Dos Santos
  Lead: *vacio*
  Fecha: *vacio*
  Enlace:https://www.larazon.es/comunidad-valenciana/rebeca-torro-nueva-secretaria-organizacion-psoe-adjudico-mas-35-millones-empresas_202507046867b95829f2a108e97e0d98.html

- **4º Documento**
  Título: *Quién es Rebeca Torró, la nueva secretaria de Organización del PSOE que sustituirá a Santos Cerdán*
  Autor: *vacio*
  Lead: *vacio*
  Fecha: *vacio*
  Enlace:https://www.larazon.es/espana/quien-rebeca-torro-nueva-secretaria-organizacion-psoe-que-sustituira-santos-cerdan_20250704686775a229f2a108e97dc5a2.html

Análisis de Documentos

Al revisar los documentos, hemos identificado varias inconsistencias en la extracción de datos, especialmente en relación con la Noticia 2.

- **Documento 2:** Este documento contiene información de la noticia que precede inmediatamente a la Noticia 2.
- **Documento 3:** Aunque este documento incluye la **URL de la Noticia 2**, hemos encontrado errores en la extracción del **título** y el **autor**. Además, no se logró extraer el **lead** ni la **fecha** de esta noticia.
- **Documento 4:** Los datos de este documento corresponden a la noticia que sigue inmediatamente a la Noticia 2.

Motivo

La información de la noticia 2 está distribuida en diferentes chunks con estructuras diferentes que no concuerdan completamente con las indicaciones del prompt. Esto provoca que el modelo no pueda extraer de manera exacta todos los datos y confunda los campos.

Explicación

Como podemos observar en los chunks, la URL de la **noticia 2** está repetida en el **chunk 36** y en el **chunk 37**. Pero el **título** y el **lead** solamente se encuentran en el **chunk 37**. Por otro lado, el autor de la **noticia 2**, y de toda la sección, se encuentra en el **chunk 39**

Chunk35

```
---Chunk-35----

[PSOE ### [Las tribulaciones de Sánchez ante su sudoku más
endiablado](https://www.larazon.es/espana/tribulaciones-sanchez-sudoku-
mas-endiablado_202507046866c7655e724b62f0684815.html "Las tribulaciones
de Sánchez ante su sudoku más endiablado") José Ignacio
Wert ### [Y si el PSOE (y los socios) de repente hacen
algo](https://www.larazon.es/espana/psoe-socios-repente-hacen-algo_2025
070468671da15e724b62f0687841.html "Y si el PSOE ylossocios de repente hacen
algo")],
```

Chunk36

```
---Chunk-36----

[[Christian Dos Santos](https://www.larazon.es/autores/christian-dos-santos/)
[](https://www.larazon.es/comunidad-valenciana/rebeca-torro-nueva-
secretaria-organizacion-psoe-adjudico-
mas-35-millones-empresas_202507046867
b95829f2a108e97e0d98.html) Cambios en el PSOE],
```

Chunk37

```
---Chunk-37----

[Cambios en el PSOE ## [Estas son las obras que la nueva secretaria de
Organización del PSOE adjudicó a una de las empresas de la trama Santos
Cerdán](https://www.larazon.es/comunidad-valenciana/rebeca-torro-nueva-
secretaria-organizacion-psoe-adjudico-mas-35-millones-empresas_20250704
6867b95829f2a108e97e0d98.html) Levantina de Ingeniería y Construcción
SL logró, en UTE con otra empresa, la ejecución de cuatro pasarelas
ciclopeatonales y en solitario la obra del Puerto de Santa Pola Perfil],
```

Chunk38

```
---Chunk-38----

[Perfil ### [Quién es Rebeca Torró, la nueva secretaria de Organización del PSOE que sustituirá a Santos Cerdán](https://www.larazon.es/espana/quien-rebeca-torro-nueva-secretaria-organizacion-psoe-que-sustituira-santos-cerdan_20250704686775a229f2a108e97dc5a2.html "Quién es Rebeca Torró, la nueva secretaria de Organización del PSOE que sustituirá a Santos Cerdán") Investigación],
```

Chunk39

```
---Chunk-39----

[Investigación ### [Un técnico se enfrentó a la trama Cerdán y no firmó las
licitaciones](https://www.larazon.es/espana/tecnico-enfrento-trama-cerdan-
firmo-licitaciones_20250704686
71ca629f2a108e97d9ed2.html Ün técnico se enfrentó a la
) [Alicia Martí](https://www.larazon.es/autores/amarti/)
[](https://www.larazon.es/espana/empresario-investigado-supuestas-
mordidas-que-pidio-hablar-directamente-santos-
cerdan-era-ultimo-que-quedaba-ver_20250704686
7c77529f2a108e97e21eb.html) Declaraciones],
```

4.3.3. El Mundo

En esta sección analizaremos la portada del 2025/06/17 del periódico El Mundo, detallando las noticias cuyos datos no se han extraído correctamente.

Figura 4.19: El Mundo 17/06/2025

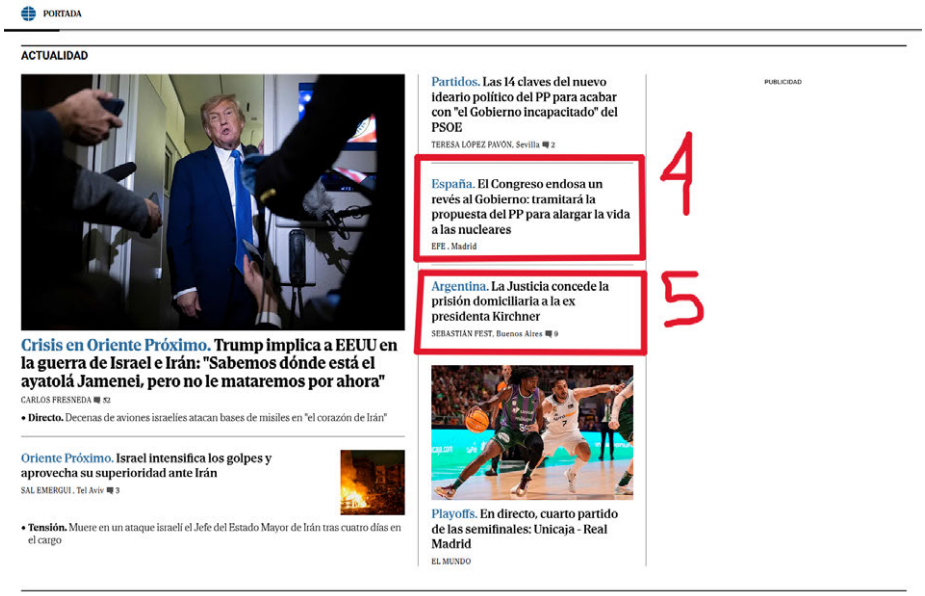


Figura 4.20: El Mundo 17/06/2025

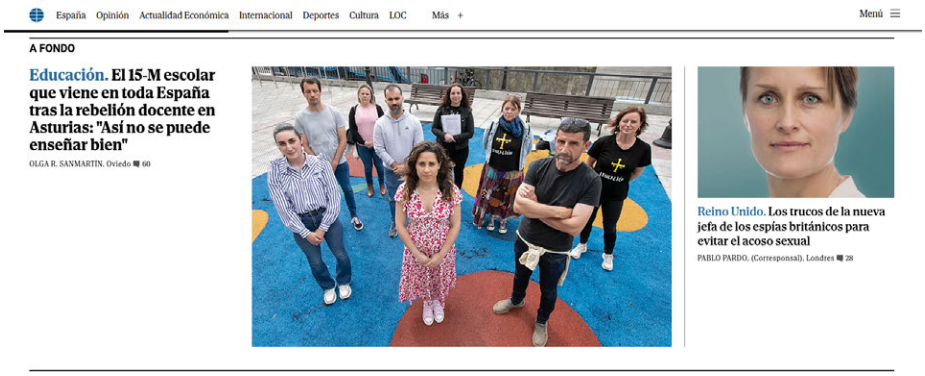


Figura 4.21: El Mundo 17/06/2025

Figura 4.22: El Mundo 17/06/2025

4.3.3.1. Primer grupo (figura 4.19)

Empezaremos analizando el grupo de noticias:
1. ‘‘El PSOE manda a sus diputados a calmar la ‘‘rabia’’ de sus militantes por el ’caso Cerdán’ aunque les ‘‘chillen’’: ‘‘Hay que preservar el Gobierno y el partido.’’”
2. ‘‘El Gobierno señala a Red Eléctrica como responsable principal del gran apagón pero implica también a las empresas privadas.”
3. ‘‘El apagón provoca pérdidas de más de 100 millones en las refinerías de Repsol y Moeve.”
A continuación, se muestran los documentos de la base de datos con los datos extraídos respecto a este grupo:

- **1º Documento**

Título: *PSOE JUDICIALIZADO*
Autor: MARTA BELVER
Lead: *vacio*
Fecha: 2025-06-17
Enlace:`https://www.elmundo.es/espana/2025/06/17/6851bb2221efa0a61a8b45a0.html`

- **2º Documento**
  Título: *Energía. El apagón provoca pérdidas de más de 100 millones en las refinerías de Repsol y Moeve*
  Autor: PAULA MARÍA
  Lead: *vacio*
  Fecha: 2025-06-15
  Enlace:`https://www.elmundo.es/economia/empresas/2025/06/15/684963c021efa0de428b4583.html`

- **3º Documento**
  Título: *El Gobierno señala a Red Eléctrica como responsable principal del gran apagón pero implica también a las empresas privadas*
  Autor: PABLO PARDO
  Lead: *vacio*
  Fecha: 2025-06-17
  Enlace:`https://www.elmundo.es/economia/empresas/2025/06/17/68514aa5fdddffc4578b456f.html`

Análisis de Documentos

Al revisar los documentos, vemos algunos errores en la extracción.

- **Documento 1:** Este documento hace referencia a la **noticia 1** con todos los campos correctamente extraidos menos el **título** incorrecto.
- **Documento 2:** Este documento contiene correctamente todos los datos de la **noticia 2**.
- **Documento 3:** Este documento contiene correctamente todos los datos de la **noticia 3**, menos el **autor** que pertenece a otra noticia.

Motivo

En este caso destacamos dos puntos importantes. El primero es que el modelo no consigue identificar el título correctamente de la noticia 1 porque la estructura del chunk no es como se esperaba. En vez de aparecer el título de primeras, hay otro elemento. El segundo caso es la duplicación de datos en los chunks, lo que provoca errores en la extracción.

Explicación

Como podemos observar en la siguiente representación de los chunks, todos los datos de **noticia 1** se encuentran en el **chunk 46** pero el LLM confunde el título de la sección de la noticia, `PSOE JUDICIALIZADO`, con el título de la noticia original. Por otro lado, el **chunk 50** contiene todos los datos de la **noticia 3**, menos el autor que se encuentra en el **chunk 51** junto a los datos de la **noticia 2** y en el **chunk 52** está duplicada la información de la **noticia 3**.

Chunk46

```
---Chunk-46----

[PSOE JUDICIALIZADO Política. ### [El PSOE manda a sus diputados
a calmar la rabia de sus militantes por el caso Cerdán
aunque les chillen: Hay que preservar el Gobierno y el
partido](https://www.elmundo.es/espana/2025/06/17/6851bb2221efa0a61a8b4
5a0.html) Redacción: [MARTA BELVER](https://www.elmundo.es/autor/marta
-belver.html)
Madrid,(https://www.elmundo.es/espana/2025/06/17/6851bb2221efa0a61a8b4
5a0.html#ancla_comentarios)][32 comentarios],
```

Chunk50

```
---Chunk-50----

[[El PSOE se queda solo tratando de impedir que Sánchez se explique
ya en el Congreso: el Gobierno quiere posponer al 9 de julio su
comparecencia](https://www.elmundo.es/espana/2025/06/17/68514ebee4d4d8c328
8b45b7.html)
Energía. ### [El Gobierno señala a Red Eléctrica como responsable
principal del gran apagón pero implica también a las empresas
privadas](https://www.elmundo.es/economia/empresas/2025/06/17/68514aa5fddd
ffc4578b456f.html)],
```

Chunk51

```
---Chunk-51----

[ Redacción: [PAULA MARÍA](https://www.elmundo.es/autor/paula-maria.html)
Madrid  Redacción: FOTO: JAVIER BARBANCHO
(https://www.elmundo.es/economia/empresas/2025/06/17/68514aa5fddd
ffc4578b456f.html#ancla_comentarios)[183 comentarios]
[Energía. El apagón provoca pérdidas de más de
100 millones en las refinerías de Repsol y
Moeve](https://www.elmundo.es/economia/empresas/2025/06/15/684963c021
efa0de428b4583.html)],
```

Chunk52

```
---Chunk-52----

[[El Gobierno señala a Red Eléctrica como responsable
principal del gran apagón pero implica también a las empresas
privadas](https://www.elmundo.es/economia/empresas/2025/06/17/68514aa5fddd
ffc4578b456f.html)
Medios. ### [El diario londinense 'Times' pide la
dimisión de Sánchez, al que acusa de no tener
principios](https://www.elmundo.es/espana/2025/06/17/68517f7b21efa0851a8
b45ac.html)
 Redacción: [PABLO PARDO](https://www.elmundo.es/autor/pablo-pardo.html)
(Corresponsal)Londres],
```

4.3.3.2. Segundo grupo (ver figura 4.20)

Para este segundo grupo nos centraremos en las noticias **4. ‘‘El Congreso endosa un revés al Gobierno: tramitará la propuesta del PP para alargar la vida a las nucleares”5.‘‘La Justicia concede la prisión domiciliaria a la ex presidenta Kirchner”**. Los documentos con los datos extraídos son:

- **4º Documento**
  Título: *El Congreso endosa un revés al Gobierno: tramitará la propuesta del PP para alargar la vida a las nucleares*
  Autor: *vacio*
  Lead: EFE Madrid
  Fecha: 2025-06-17
  Enlace:`https://www.elmundo.es/espana/2025/06/17/6851c520e4d4d89f3e8b456f.html`

- **5º Documento**
  Título: *La Justicia concede la prisión domiciliaria a la ex presidenta Kirchner*
  Autor: *vacio*
  Lead: *vacio*
  Fecha: 2025-06-17
  Enlace:`https://www.elmundo.es/internacional/2025/06/17/6851b660fdddffb1518b457a.html`

Análisis de Documentos

- **Documento 4:** Este documento corresponde a la **noticia 4** y se ha extraido mal el **autor** y **lead**, se han intercambiado.
- **Documento 5:**Este documento corresponde a la **noticia 5** y tiene los campos correctamente extraídos, menos el **autor**

Motivo

Aquí se manifiesta el problema de tener información duplicada. Esto se debe a la estructura del código de la web, por cómo está diseñada esta, la extracción te devuelve datos duplicados. Esto provoca confusión en el modelo, añadido a su baja capacidad, haciéndolo incapaz de extraer todo correctamente.

Explicación

Como se puede ver, el **chunk 61** contiene toda la información de la **noticia 4** y esta misma información está repetida en el **chunk 62**. Aún así, el modelo no es capaz de extraer correctamente el autor, en este caso EFE Madrid. El **chunk 62** también contiene la información de la **noticia 5**, que también esta información está repetida en el **chunk 63**. El modelo no consigue obtener el autor, Sebastián Fest en este caso.

Chunk61

```
---Chunk-61----

[[Las 14 claves del nuevo ideario político del
PP para acabar con 'el Gobierno incapacitado' del
PSOE](https://www.elmundo.es/espana/2025/06/17/6851ab61fdddffdd5a8b4591.html)
España. ### [El Congreso endosa un revés al Gobierno:
tramitará la propuesta del PP para alargar la vida a las
nucleares](https://www.elmundo.es/espana/2025/06/17/6851c520e4d4d89f3e8
b456f.html)
 Redacción: EFE Madrid],
```

Chunk62

```
---Chunk-62----

 Redacción: EFE Madrid
El Congreso endosa un revés al Gobierno: tramitará la
propuesta del PP para alargar la vida a las nucleares,
(https://www.elmundo.es/espana/2025/06/17/6851c520e4d4d89f3e8b456f.html)
Argentina. ### [La Justicia concede la
prisión domiciliaria a la ex presidenta
Kirchner](https://www.elmundo.es/internacional/2025/06/17/6851b660fdddffb
1518b457a.html)
 Redacción: [SEBASTIÁN FEST](https://www.elmundo.es/autor/el-mundo.html)
Buenos Aires],
```

Chunk63

```
---Chunk-63----

(https://www.elmundo.es/internacional/2025/06/17/6851b660fddd
ffb1518b457a.html#ancla_comentarios)[9 comentarios]
La Justicia concede la prisión domiciliaria a la ex presidenta Kirchner
,(https://www.elmundo.es/internacional/2025/06/17/6851b660fdddffb1518b457a.html)
Playoffs.       ###       [En       directo,       cuarto       partido
de        las        semifinales:        Unicaja        -        Real
Madrid](https://www.elmundo.es/deportes/baloncesto/liga-endesa/unicaja
-real-madrid/2025/06/17
/02_0201_20250617_6203_6210-directo.html),
```

4.3.3.3. Tercer Grupo (ver figura 4.22)

En este caso nos centraremos en las noticias:
6. ‘‘Santiago Cañizares: ‘‘Subastaron la casa de mis padres y salí adelante con dinero prestado de Txetxu Rojo’’”
7.‘‘Dimite el presidente del TEAC y número tres de Montero tras ser acusado de cobrar de empresarios para retirarles sanciones ”

- **6º Documento**
  Título: *Santiago Cañizares: ”Subastaron la casa de mis padres y salí adelante con dinero prestado de Txetxu Rojo”*
  Autor: *vacio*
  Lead: *vacio*
  Fecha: *vacio*
  Enlace:`https://www.elmundo.es/papel/lideres/2025/06/17/6849aefe21efa04e518b4591.html`

- **7º Documento**
  Título: *Dimite el presidente del TEAC y nÃºmero tres de Montero tras ser acusado de cobrar de empresarios para retirarles sanciones*
  Autor: *vacio*
  Lead: *vacio*
  Fecha: *vacio*
  Enlace:`https://www.elmundo.es/economia/2025/06/17/685148b2fc6c83345e8b4594.html`

Análisis de Documentos

- **Documento 6:** Contiene los datos correctamente extraidos de la **noticia 6**, menos del **autor** y la **fecha**

- **Documento 7:** Contiene los datos extraidos de la **noticia 7** pero los campos **título, autor y fecha** no se han extraido correctamente.

Motivo

Además de que la información de las noticias está dividida y parte repetida en diferentes chunks, en este ejemplo se puede observar cómo la obtención de código a Markdown tiene errores de codificación de caracteres. Por eso, en la noticia 7 aparece el título con caracteres raros. Sin embargo, a veces, como en la noticia 6, el modelo es capaz de corregir estos errores.

4.3.4. Explicación

Podemos observar cómo en todos los chunks existe un error de codificación de caracteres, debido a un problema con la extracción del código de la web. El **chunk 68** contiene la información de la **noticia 6**, menos el autor que está en el **chunk 69** donde también se repite información de la noticia. En el **chunk 72** podemos observar que está la información de la **noticia 7**, que también se repite en el **chunk 73**.

Chunk68

```
---Chunk-68----

[[28 comentarios]Los trucos de la nueva jefa de los espÃas britÃ¡nicos
para evitar el acoso sexual
AL DÃA
Entrevista no vista. ### [Santiago CaÃ±izares: "Subastaron la
casa de mis padres y salÃ adelante con dinero prestado de Txetxu
Rojo"](https://www.elmundo.es/papel/lideres/2025/06/17/6849aefe21efa04e518
b4591.html)],
```

Chunk69

```
---Chunk-69----

[ RedacciÃ³n: [DAVID LEMA](https://www.elmundo.es/autor/david-lema.html)
(https://www.elmundo.es/papel/lideres/2025/06/17/6849aefe21efa04e518b45
91.html#ancla_comentarios)[13 comentarios]
Santiago CaÃ±izares: "Subastaron la casa de mis padres y salÃ adelante
con dinero prestado de Txetxu Rojo"[URL]
```

Chunk72

```
---Chunk-72----

[[Balas rusas y georgianas contra Grinda: las tres
'muertes' del fiscal en el punto de mira de la 'comisiÃ³n
Leire'](https://www.elmundo.es/cronica/2025/06/17/684d03c0e4d4d8ea678b45
7f.html)
EconomÃa. ### [Dimite el presidente del TEAC y nÃºmero tres de
Montero tras ser acusado de cobrar de empresarios para retirarles
sanciones](https://www.elmundo.es/economia/2025/06/17/685148b2fc6c83345e8b
4594.html)
 RedacciÃ³n: [ALEJANDRA OLCESE](https://www.elmundo.es/autor/alejandra
-olcese.html)],
```

Chunk73

```
---Chunk-73----

[[73 comentarios](https://www.elmundo.es/economia/2025/06/17/685148b2fc6
c83345e8b4594.html#ancla_comentarios)
Dimite el presidente del TEAC y nÃºmero tres de Montero tras ser acusado
de cobrar de empresarios para retirarles sanciones[URL],
```

4.3.5. ABC

En este apartado analizaremos la portada del 2025/07/04 del periódico ABC destacando las noticias donde los datos no se han extraído correctamente.

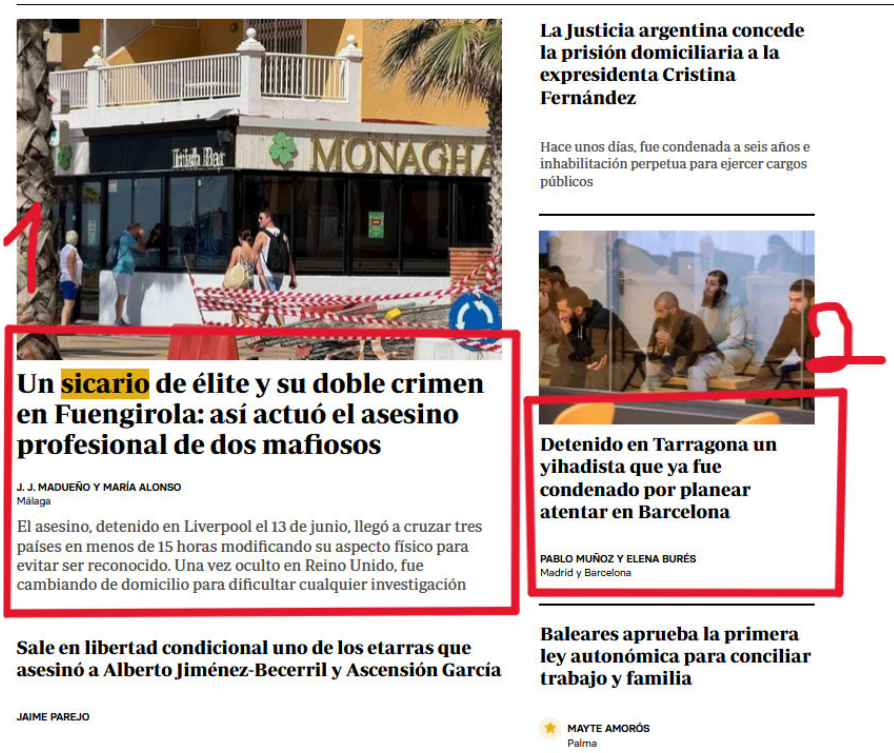


Figura 4.23: ABC 04/07/2025

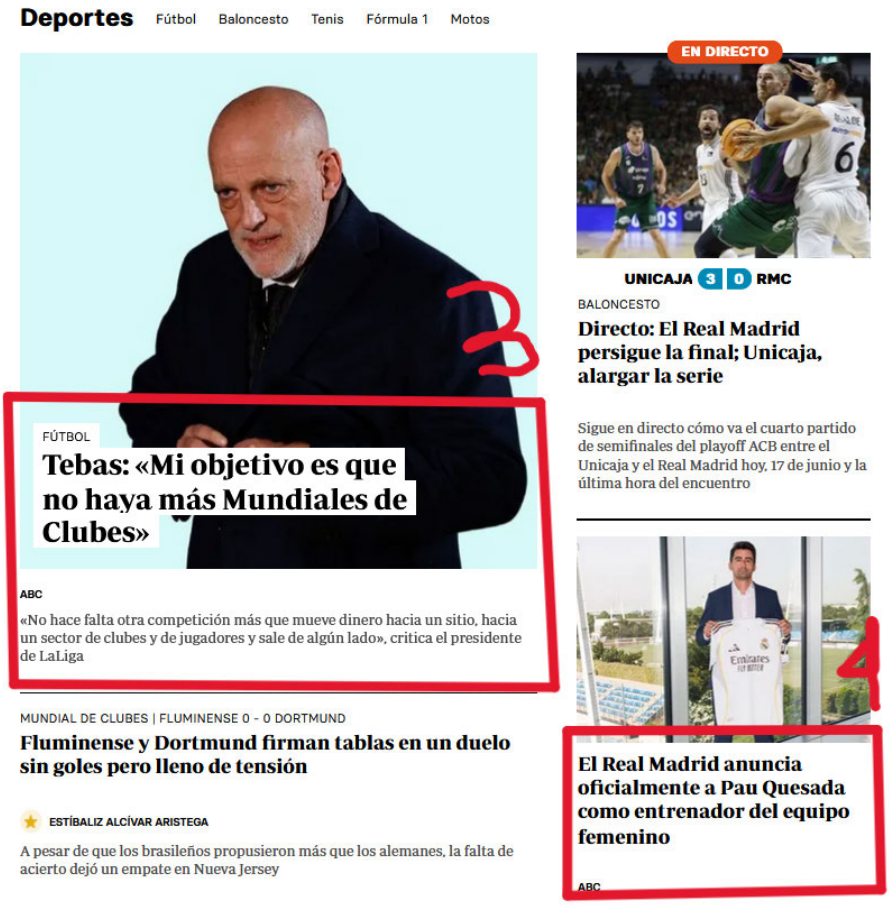


Figura 4.24: ABC 04/07/2025

4.3.5.1. Primer grupo (ver figura 4.23)

Las noticias que vamos a analizar son **1. ‘‘Un sicario de élite y su doble crimen en Fuengirola: así actuó el asesino profesional de dos mafiosos”. 2. ‘‘Detenido en Tarragona un yihadista que ya fue condenado por planear atentar en Barcelona”** Los documentos con los datos extraídos son los siguientes:

- **1º Documento**
  Título: *Un sicario de élite y su doble crimen en Fuengirola: así actuó el asesino profesional de dos mafiosos*
  Autor: J. J. Madueño y María Alonso Málaga
  Lead: *vacio*
  Fecha: 2025-06-17
  Enlace:https://www.abc.es/espana/andalucia/malaga/sicario-elite-doble-asesinato-fuengirola-actuo-asesino-20250617133815-nts.html

- **2º Documento**

Título: *Detenido en Tarragona un yihadista que ya fue condenado por planear atentar en Barcelona*
Autor: Pablo Muñoz y Elena Burés
Lead: *vacio*
Fecha: *vacio*
Enlace:https://www.abc.es/espana/cataluna/detenido-tarragona-yihadista-condenado-planear-atentar-barcelona-20250617172516-nt.html

Análisis de Documentos

- **Documento 1:** Este documento hace referencia a la **noticia 1** con todos los campos correctamente extraidos menos el **lead** que no ha podido extraerlo.
- **Documento 2:** Este documento contiene correctamente todos los datos de la **noticia 2**, menos la **fecha** que no ha podido extraerla.

Motivo

El motivo principal de que no se haya extraído el campo lead de la noticia 1 es la división de la información en diferentes chunks. El modelo no es capaz de ver la relación entre los datos de un chunk y otro. Por otro lado, el modelo no ha sido capaz de identificar el patrón de la fecha dentro de la URL en este caso.

Explicación

Como se puede observar, el **chunk 67** contiene los datos de la **noticia 1**, menos el campo **lead** que se encuentra en el **chunk 68**. Por otro lado, los **chunks 71** y **72** contienen todos los datos de la **noticia 2**, pero el modelo no ha sido capaz de extraer la fecha. A continuación se representan los chunks implicados:

Chunk67

```
---Chunk-67----

[José Manuel Nieves
## [ Un sicario de élite y su doble crimen en
Fuengirola: así actuó el asesino profesional de dos
mafiosos](https://www.abc.es/espana/andalucia/malaga/sicario-elite-doble
-asesinato-fuengirola-actuo-asesino-20250617133815-nts.html Ün sicario
de élite y su doble crimen en Fuengirola: así actuó el asesino profesional
de dos mafiosos”)
J. J. Madueño y María Alonso Málaga],
```

Chunk68

```
---Chunk-68----

[J. J. Madueño y María Alonso Málaga
El asesino, detenido en Liverpool el 13 de junio, llegó a cruzar tres
países en menos de 15 horas modificando su aspecto físico para evitar ser
reconocido. Una vez oculto en Reino Unido, fue cambiando de domicilio para
dificultar cualquier investigación],
```

Chunk71

```
---Chunk-71----

[## [ Detenido en Tarragona un yihadista
que ya fue condenado por planear atentar en
Barcelona](https://www.abc.es/espana/cataluna/detenido-tarragona
-yihadista-condenado-planear-atentar-barcelona-20250617172516-nt.html
"Detenido en Tarragona un yihadista que ya fue condenado por planear
atentar en Barcelona")
Pablo Muñoz y Elena Burés Madrid y Barcelona],
```

Chunk72

```
---Chunk-72----

[Pablo Muñoz y Elena Burés Madrid y Barcelona
## [ Baleares aprueba la primera ley autonómica para conciliar trabajo y
familia](https://www.abc.es/espana/baleares/baleares-aprueba-primera
-ley-autonomica-conciliar-trabajo-20250617181022-nt.html "Baleares
aprueba la primera ley autonómica para conciliar trabajo y familia")
Mayte Amorós Palma
Ignacio Camacho
## [ El santismo](https://www.abc.es/opinion/ignacio-camacho-santismo
-20250617190744-nt.html 'El santismo')],
```

4.3.5.2. Segundo grupo (ver figura 4.24)

Las noticias que detallaremos son: **1. ‘‘Tebas: ≪Mi objetivo es que no haya más Mundiales de Clubes≫”. 2.‘‘El Real Madrid anuncia oficialmente a Pau Quesada como entrenador del equipo femenino”**. Los documentos extraídos de la base de datos en referencia a estas noticias son :

- **3º Documento**
  Título: *Tebas: ≪Mi objetivo es que no haya más Mundiales de Clubes≫*
  Autor: *vacio*
  Lead: *vacio*
  Fecha: *vacio*
  Enlace:`https://www.abc.es/deportes/futbol/tebas-objetivo-mundiales-clubes-20250617142802-nt.html`

- **4º Documento**

Título: *El Real Madrid anuncia oficialmente a Pau Quesada como entrenador del equipo femenino*
Autor: *vacio*
Lead: *vacio*
Fecha: *vacio*
Enlace:`https://www.abc.es/deportes/futbol/real-madrid-anuncia-oficialmente-pau-quesada-entrenador-20250617155641-nt.html`

Análisis de Documentos

- **Documento 3:** Este documento hace referencia a la **noticia 3** y no se ha podido extraer correctamente loa campos de **fecha, autor y lead**.
- **Documento 4:** Este documento contiene los datos de la **noticia 4**, pero los campos **fecha y autor** no se han extraido bien.

Motivo

La problemática se repite como en el caso anterior. La información de una misma noticia está dividida en dos chunks y el modelo no es capaz de relacionarla, como es en el caso de la extracción del campo lead. Además, el modelo no es capaz de identificar y extraer el patrón del campo de fecha y tampoco ha sido capaz de reconocer el autor, que en este caso es ABC.

Explicación

Es este caso el **chunk 91** contiene los datos de la **noticia 3**, menos el campo **lead** que se encuentra en el siguiente, el **chunk 92**. Por otro lado, el **chunk 95** contiene todos los datos de la **noticia 4**, pero el modelo no es capaz de extraer los campos **autor** ni **fecha**. Los chunks implicados son:

Chunk91

```
---Chunk-91----

[* [ Tenis ](https://www.abc.es/deportes/tenis/)
* [ Fórmula 1 ](https://www.abc.es/deportes/formula-1/)
* [ Motos ](https://www.abc.es/deportes/motos/)
Fútbol
## [ Tebas: ≪Mi objetivo es que no haya más Mundiales de
Clubes≫](https://www.abc.es/deportes/futbol/tebas-objetivo-mundiales
-clubes-20250617142802-nt.html "Tebas: ≪Mi objetivo es que no haya más
Mundiales de Clubes≫")
ABC],
```

Chunk92

```
---Chunk-92----

[ABC
≪No hace falta otra competición más que mueve dinero hacia un sitio,
hacia un sector de clubes y de jugadores y sale de algún lado≫, critica
el presidente de LaLiga
MUNDIAL DE CLUBES | fluminense 0 - 0 dortmund],
```

Chunk95

```
---Chunk-95----

[## [ El Real Madrid anuncia oficialmente a Pau Quesada como entrenador del
equipo femenino](https://www.abc.es/deportes/futbol/real-madrid-anuncia
-oficialmente-pau-quesada-entrenador-20250617155641-nt.html 'El Real
Madrid anuncia oficialmente a Pau Quesada como entrenador del equipo
femenino')
ABC],
```

4.4. Objetivos logrados

Empezaremos razonando por los objetivos específicos. **Objetivos específicos:**

1. **Desarrollar un sistema de gestión de tareas:** Este sistema ha sido completado satisfactoriamente gracias al uso de **Celery**, para ejecución en segundo plano de los procesos, y **RedBeat**, para la programación dinámica de los procesos, junto con **Redis** como backend para el correcto funcionamiento.

2. **Implementar una interfaz atractiva:** Se ha completado creando un frontend claro y sencillo, con pantallas dedicadas a crear tareas, listar tareas expiradas, listar taread programadas y consultar los datos extraídos. El diseño facilita la experiencia de usuario utilizando componentes simple de la libreria **Streamlit**.

3. **Implementar una REST API consistente:** Gracias a **FastAPI** se han definido todos los endpoints necesarios para la comunicación con todas las funcionalidades de la aplicación.

4. **Crear una base de datos no relacional:** Se ha creado satisfactoriamente una base datos usando **MongoDB** para almacenar en una única colección todos los documentos extraídos. Con campos los campos: título, autor, lead, URL, fecha de publicación y task name, permitiendo un acceso eficiente a la información obtenida por cada proceso.

5. **Diseñar una arquitectura de miscroservicios:** Este objetivo se ha conseguido completar gracias al uso de **Docker**. Se han encapsulado los servicios en contenedores independientes, el frontend(Streamlit), el

backend (FastAPI), el scheduler (RedBeat), los workers (Celery), Redis y MongoDB, orquestando su despliegue y su comunicación eficiente.

6. **Implementar un ensayo con medios específicos:** Hemos podido completar este objetivo construyendo el MVP (Minimum Viable Product), validando los procesos en un caso real con los periódicos El País, La Razón, El Mundo y ABC.

4.5. Problemas encontrados

Durante el desarrollo del proyecto **NewScrAIper** se han identificado diversos problemas. A continuación, se describen los más relevantes:

- **Limitaciones del modelo LLM**
  Aunque el modelo LLaMa-3 8B ha demostrado un rendimiento notable en la identificación de los campos deseados, en ciertos casos donde la estructura que analiza es muy caótica, el modelo ha generado resultados inconsistentes (como confundir los campos o no reconocer los patrones de la fecha de publicación). Esto en parte es por la baja capacidad del modelo, al ser solamente de 8 billones (anglosajones) de parámetros.

  La limitación más importante es la limitación del hardware. Para este proyecto se ha usado un modelo que era posible correr en un ordenador "normal"(con un procesador Intel i5 y una gráfica Nvidia GTX 1050). Esto afecta directamente a la eficacia y eficiencia de la aplicación. Con un ordenador más potente se podría correr un modelo más grande y más capaz mejorando la aplicación.

  Otra limitación es la ventana de contexto del modelo, que hace referencia a la cantidad de tokens que un modelo puede recordar. Esto obliga a añadir un poceso de división antes del proceso de extracción, con lo que puede conllevar a algunos fallos.

- **División de *chunks* en la entrada del modelo**
  Debido al limite de la ventana de contexto, fue necesario dividir el contenido en fragmentos (*chunks*) antes de enviarlo al LLM. Esta división no siempre respeta la longitud de los artículos, lo que provoca algunos errores como pérdida de contexto o información duplicada

- **Gestión del `task_id` en tareas programadas**
  `task_id` de las tareas ejecutadas por RedBeat a través de Celery. Al ser tareas programadas automáticamente, el `task_id` solo se conoce una vez se ejecuta realmente la tarea, lo que ha obligado a instrumentar código adicional para interceptar y almacenar dicho identificador durante la ejecución para su posterior trazabilidad en MongoDB.

- **Inestabilidad de Crawl4ai**
  Debido a la juventud de la libreria Crawl4ai, existen algunas inconsistencias. Provacando que a veces el proceso de obtención del código no se ejecute correctamente y el proceso de extracción no tenga nada que

extraer. Otro problema destacable es mala codificación de caracteres en algunos casos.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

Este proyecto aborda la necesidad de desarrollar un método general para extraer datos de noticias de varios periódicos digitales, un proceso que, por lo general, se realiza mediante un enfoque específico para cada caso. Para lograr este objetivo, nos enfocamos en usar los modelos de lenguaje de gran escala, o **LLM** (Large Language Models), por su capacidad para comprender y generar lenguaje humano, permitiendo así una solución más flexible y adaptable. Para demostrar dichas capacidades, se ha desarrollado la aplicación **NewScrAIper**[1].

NewScrAIper es una aplicación web diseñada para generalizar y automatizar el proceso de extracción de noticias de los principales periódicos digitales españoles. La aplicación integra un **módulo de web scraping adaptativo**, utilizando **Crawl4AI** para obtener el código fuente web y la librería **Langchain** con el modelo LLaMa-3 8B, en local, para ejecutar el proceso de extracción de los datos: título, autor, fecha, lead y URL de la noticia original. Estas herramientas demuestran que existe una ventaja al emplear LLM para generalizar el proceso de web scraping de noticias.

A nivel arquitectónico, se ha diseñado una **infraestructura de microservicios** utilizando contenedores Docker, capaz de operar de forma ininterrumpida (24/7) y de escalar horizontalmente según la demanda. Un **sistema de gestión de tareas** mediante **Celery** y **RedBeat** sobre Redis, posibilitando la programación dinámica de las tareas, según lo demande el usuario, y ejecutando dichas tareas de web scraping en segundo plano. Además, se ha construido una interfaz de usuario atractiva y funcional, usando **Streamlit**, con la que los usuarios pueden crear, listar y gestionar las tareas de web scraping, así como visualizar y acceder a los datos extraídos.

Complementariamente se ha desarrollado un sólido backend, creando una **REST API** con **FastAPI**, con todos los endpoints necesarios para interactuar con el sistema (creación, listado, cancelación y recuperación de resultados) asegurando una correcta comunicación. La información extraída se almacena en una base de datos no relacional, usando **MongoDB**, diseñada para manejar grandes volúmenes de datos, donde cada elemento representa los datos extraídos de un artículo de noticia, incluyendo título, autor, lead, URL, fecha y el nombre de la tarea. Este enfoque arquitectónico no solo garantiza la independencia y escalabilidad de los servicios, sino que también facilita el mantenimiento del sistema.

Se implementó un ensayo de extracción de noticias y su procesamiento para

[1] github.com/ec-dani/newscraiper

los medios de El País, El Mundo, ABC y La Razón, validando la funcionalidad del sistema en un escenario real. Los resultados muestran mejoras en adaptabilidad y mantenibilidad .

Sin embargo, también ponen de manifiesto áreas de mejora, como en la división de chunks y las limitaciones del modelo LLaMa-3 8B, que, al ser de menor tamaño, muestra algunas dificultades para extraer la información en comparación con modelos más potentes. Asimismo, se plantea mejorar el sistema creando un componente para visualizar el estado de las tareas de extracción en curso, lo que facilitaría un mejor control y transparencia del proceso. Además, se incluirá el mecanismo de filtrado por keywords que permita filtrar las noticias.

NewScrAIper abre la puerta a una nueva generación de herramientas de extracción de datos web, más generales y sostenibles.

5.2. Lineas futuras

A continuación, detallamos los puntos que se pueden mejorar:

1. **Módulo de web scraping adaptativo**: Se ha desarrollado un componente híbrido que integra Crawl4AI para la obtención del HTML y LangChain para la identificación y extracción de los campos deseados (título, autor, fecha, lead y URL) de cualquier periódico digital, utilizando un LLM. Sin embargo, se han observan varios errores en los resultados por lo que no se puede confiar al cien por cien.

   Para aumentar la efectividad de este módulo, es necesario optimizar la división del Markdown del código de la página, ajustar el modelo para este problema en concreto (fine-tuning) o elegir un LLM con más parámetros. Dependiendo de la circunstancia, se pueden implementar estas soluciones de forma individual o conjuntamente.

2. **Filtrado por Keywords:** No se ha desarrollado este punto en el MVP. Una posible solución es usar la misma libreria **Crawl4ai** para que filtre el contenido antes de generar el Mardown. Otra solución sería que antes de insertar la noticia, compruebe que está relacionana con las keywords.

3. **Monitoreo a tiempo real:** Para mejorar la monitorización de la aplicación, se puede añadir una ventana donde el usuario consulte las tareas que se estan ejecutando en ese momento.

   Una solucion puede ser ligar la tarea programa (de **RedBeat**) con la tarea de ejecución (de **Celery**) y guardar esa relación la en una base de datos (como **Redis**).

5.3. Impacto social y medioambiental

La adopción masiva de modelos de lenguaje de gran escala plantea un desafío de sostenibilidad debido al alto consumo de energía y emisiones de carbono. Es

esencial adoptar prácticas más responsables para aprovechar sus beneficios sin dañar el planeta. El impacto ambiental de los LLM es cada vez más evidente. Tanto su entrenamiento como su operación requieren enormes recursos energéticos. Por ejemplo, GPT-3 consumió alrededor de 1 GWh de electricidad por día durante su entrenamiento [9] , una cantidad comparable a la demanda eléctrica de una pequeña ciudad. Del mismo modo, se ha visto que desarrollar modelos de 7 y 13 mil millones de parámetros puede generar 159 tCO2eq emitidas, equivalente a 843 kL de agua consumida [10]. Estos requerimientos generan un uso energético sustancial que se traduce en un aumento significativo de la huella de carbono global.

Una vez creados, es importante ver también el consumo por consulta. Aunque no hay datos concretos para el consumo exacto del modelo LLaMa3-8B, se estima que el gasto energético por consulta es de 0,0158 Wh (57 julios [2]), una estimación de emisiones de Co2 equivalente a 17g [10] y un consumo de agua de 760ml [10]. En nuestro caso, el proceso de extracción de datos hace un uso medio del LLM de una hora y media con 170 consultas.

- **Consumo energético estimado:**

$$170 \text{ consultas} \times 0{,}0158\,\text{Wh} = 2{,}686\,\text{Wh} \approx 2{,}69\,\text{Wh}$$

- **Emisiones de CO_2 equivalentes:**

$$170 \text{ consultas} \times 17\,\text{g} = 2890\,\text{g } CO_2 = 2{,}89\,\text{kg } CO_2$$

- **Consumo de agua estimado:**

$$170 \text{ consultas} \times 760\,\text{ml} = 129{,}200\,\text{ml} = 129{,}2\,\text{litros}$$

Este consumo se puede comparar a cosas cotidianas como dejar una bombilla LED de 10 W encendida durante unos 16 minutos, conducir un coche a gasolina unos 20 km, fabricar una camiseta de algodón o una ducha de 10–12 minutos.

Estos datos son meramente estimaciones, ya que no se han publicado datos concretos hasta el día de hoy.

El impacto social del web scraping puede ser positivo o negativo, dependiendo de cómo se utilice. Centrándonos en la parte de noticias, obtener un gran volúmen de datos permite identificar patrones y tendencias, revelando sesgos de los periódicos al analizar las diferentes maneras en las que tratan una misma noticia. Además, es una herramienta de gran ayuda contra la desinformación, ya que se podría identificar rápidamente contenido falso, contrastar su información y rastrear su propagación. Por otro lado, el mayor reto es el tema de los derechos de autor. Scrapear y republicar noticias sin permiso infringe la propiedad intelectual, perjudicando así a los redactores y al periódico. Desde un punto de vista más técnico, el scraping masivo puede reducir el tráfico directo a los sitios web de noticias, lo que a su vez disminuye los ingresos por publicidad y suscripciones. Para aprovechar sus beneficios y mitigar sus perjuicios, es fundamental establecer un equilibrio entre el acceso a la información y el respeto por los derechos de los creadores de contenido, promoviendo prácticas éticas.

[2]MIT Technology Review

Bibliografía

[1] J. Sixto-García and X. López-García, "Innovative innovation in journalism," *Journalism*, vol. 26, no. 1, pp. 65–88, 2025. [Online]. Available: https://doi.org/10.1177/14648849231219359

[2] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, "A comprehensive overview of large language models," 2024. [Online]. Available: https://arxiv.org/abs/2307.06435

[3] Y. Annepaka and P. Pakray, "Large language models: a survey of their development, capabilities, and applications," *Knowledge and Information Systems*, vol. 67, no. 3, pp. 2967–3022, Mar. 2025. [Online]. Available: https://doi.org/10.1007/s10115-024-02310-4

[4] D. Nam, A. Macvean, V. Hellendoorn, B. Vasilescu, and B. Myers, "Using an llm to help with code understanding," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ser. ICSE '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3597503.3639187

[5] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie, "A survey on evaluation of large language models," 2023. [Online]. Available: https://arxiv.org/abs/2307.03109

[6] H. Zhao, H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, and M. Du, "Explainability for large language models: A survey," 2023. [Online]. Available: https://arxiv.org/abs/2309.01029

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[8] S. R. Bowman, "Eight things to know about large language models," 2023. [Online]. Available: https://arxiv.org/abs/2304.00612

[9] A. Singh, N. P. Patel, A. Ehtesham, S. Kumar, and T. T. Khoei, "A survey of sustainability in large language models: Applications, economics, and challenges," 2025. [Online]. Available: https://arxiv.org/abs/2412.04782

[10] J. Morrison, C. Na, J. Fernandez, T. Dettmers, E. Strubell, and J. Dodge, "Holistically evaluating the environmental impact of creating language models," 2025. [Online]. Available: https://arxiv.org/abs/2503.05804

Anexos