Web Scraping para la centralización de Cursos de Extensión en Universidades Públicas Argentinas

Julieta Rivero; Jose Rámirez; Enzo Defranco; Rafael Eguren; Simón Koenig; Valeria Lasagna; Romina Istvan

Laboratorio de Ingeniería en Sistemas de Información, LINES UTN FRLP
Av. 60 s/n° esquina 124, La Plata, Buenos Aires, Argentina
{jrivero; jramirezchavez; enzo.defranco; rafael_eguren; simonkoenig}@alu.frlp.utn.edu.ar
{valerial; ristvan}@frlp.utn.edu.ar

Abstract. El acceso a la información sobre cursos de extensión ofrecidos por las universidades públicas argentinas se encuentra actualmente disperso en múltiples sitios institucionales, lo que dificulta su localización y aprovechamiento. Para abordar esta problemática, se presenta en este trabajo el diseño y desarrollo de una solución destinada a centralizar la oferta de cursos utilizando la técnica de Web Scraping, la cual permite la extracción automatizada y estructurada de datos desde sitios web.
En una primera etapa, el proyecto se enfoca en las Facultades Regionales de la Universidad Tecnológica Nacional (UTN). Para ello, emplea Python y BeautifulSoup para la extracción de datos, y tecnología HTML, CSS, Bootstrap y JavaScript para el desarrollo de un sitio web. Para complementar este ecosistema tecnológico, desarrolla una Progressive Web App (PWA), que proporciona una experiencia de usuario mejorada al permitir la instalación en dispositivos móviles.
Los resultados iniciales son prometedores, con la extracción exitosa de datos completos de dos facultades regionales de la UTN. A medida que el proyecto avanza, se planea ampliar la base de datos para incluir otras universidades públicas argentinas y mejorar la experiencia del usuario mediante opciones avanzadas de filtrado y la exploración de nuevas tecnologías. Se espera que estos esfuerzos maximicen el impacto en la comunidad educativa y contribuyan significativamente a la mejora del acceso a oportunidades de educación continua en Argentina.

Keywords: Web Scraping, Raspado Web, Scraping, Cursos de Extensión, Universidades Públicas Argentinas.

1 Introducción. Motivación y Contexto

El Laboratorio de Ingeniería en Sistemas de Información (LINES) de la UTN La Plata se dedica a la investigación y desarrollo en el campo de la tecnología de la información.

Dentro de este grupo de trabajo, un equipo de becarios identificó una problemática puntual y significativa en el acceso a la información sobre cursos de extensión ofrecidos por las universidades públicas argentinas. Actualmente, la información sobre estos cursos está dispersa en distintos sitios web, lo que dificulta a estudiantes,

profesionales y a la sociedad en general encontrar y aprovechar oportunidades educativas, conllevando a una pérdida de valiosas oportunidades para la educación continua.

Para abordar esta problemática, el equipo comienza con el desarrollo de una aplicación diseñada para centralizar la oferta de cursos de extensión brindados por las distintas universidades públicas argentinas. Utiliza para ello *Web Scraping* [1], una técnica que automatiza la extracción de datos de sitios web y los convierte a un formato estructurado; de esta manera, busca y extrae información disponible en los sitios, garantizando así la precisión y la actualización constante de los datos recopilados.

Este proyecto también contempla, en una segunda etapa, la ampliación de la base de conocimientos de un modelo de lenguaje grande (Large Language Model, LLM). Este modelo está en proceso de entrenamiento en el laboratorio, y es llevado a cabo por un equipo especializado en Inteligencia Artificial. Uno de sus objetivos es desarrollar un chatbot que aprenda las características de un dominio académico específico y responda preguntas de manera informativa y completa.

2 Desarrollo. Aporte del Proyecto

2.1 Investigación y análisis de técnicas de extracción de datos de un sitio web

Para consolidar, de manera efectiva, los datos sobre cursos de extensión ofrecidos por universidades públicas argentinas resulta esencial seleccionar la técnica de extracción de datos más adecuada. Esto requiere investigar y comparar los métodos disponibles para elegir el más adecuado para el proyecto:

Web Scraping (Raspado Web): posibilita la extracción automática de datos de páginas web mediante el uso de bots o crawlers. Permite obtener información estructurada o no estructurada utilizando técnicas como la inspección de elementos HTML y patrones de búsqueda [2].

API (Interfaz de Programación de Aplicaciones): son conjuntos de reglas y protocolos que permiten a diferentes aplicaciones interactuar entre sí. Algunos sitios web ofrecen APIs públicas que permiten acceder a sus datos de manera estructurada y controlada [3].

Aunque en el pasado fue una opción popular, el uso de **RSS (Really Simple Syndication)** ha disminuido en favor de APIs y Web Scraping. Los feeds RSS permiten la suscripción y obtención de actualizaciones periódicas en un formato estandarizado; sin embargo, su capacidad para acceder a datos específicos o detallados es limitada, ya que depende de los feeds ofrecidos por los sitios web. Esta restricción reduce su flexibilidad y profundidad en la recolección de datos [4].

La obtención de datos mediante Web Scraping y el uso de una API pueden ser comparados en términos de disponibilidad, robustez, complejidad de implementación y mantenimiento, y también desde una perspectiva legal y ética [2] [5] [6].

Disponibilidad: Web Scraping ofrece la ventaja de acceder a información incluso cuando no existe una API pública disponible. Sin embargo, esto requiere un manejo cuidadoso para asegurar la precisión y estabilidad de los datos extraídos.

Robustez: La confiabilidad del Web Scraping puede verse afectada por cambios en la estructura HTML del sitio o medidas anti-scraping. En contraste, las APIs brindan

acceso estructurado y controlado con documentación detallada, lo que garantiza una mayor estabilidad y consistencia.

Complejidad y Mantenimiento: Web Scraping puede implicar un desarrollo inicial más complejo y un mantenimiento continuo debido a posibles cambios en el diseño del sitio web. Las APIs, en cambio, suelen ser más sencillas de implementar y mantener, ya que están diseñadas para proporcionar datos de manera consistente y estable.

Perspectiva Legal y Ética: El Web Scraping puede enfrentar desafíos legales y éticos, como posibles infracciones de los términos de servicio de los sitios web. Por el contrario, el uso de APIs generalmente está más regulado y clarificado por las condiciones del proveedor, ofreciendo un marco legal más definido.

En conclusión, se debe dar preferencia al uso de APIs cuando estén disponibles, debido a su robustez y eficiencia en la extracción de datos. En ausencia de APIs, el Web Scraping se presenta como una alternativa válida. El uso de RSS queda relegado en este contexto debido a su limitada flexibilidad y profundidad en la recolección de datos.

2.2 Desarrollo de la aplicación

En una primera etapa del proyecto, se focaliza en las Facultades Regionales de la UTN. Se lleva a cabo una investigación detallada de la estructura de cada página web asociada a estas facultades para identificar la información disponible sobre los cursos de extensión y determinar cómo acceder a estos datos.

Luego de este análisis, se seleccionan tecnologías y herramientas de Web Scraping para la extracción y el manejo de datos desde sitios, ya que no se disponen de APIs públicas que proporcionen la información requerida.

Seguidamente se diseña y desarrolla un sitio web. Entre las tecnologías de Frontend que se utilizan, se encuentra: HTML para estructurar el contenido de la página y CSS para gestionar los estilos visuales de la página, junto con Bootstrap [7] para garantizar una vista responsiva. También se utiliza JavaScript para dar dinamismo y posibilitar la visualización de los datos obtenidos a través del scraping en el Frontend.

Para mejorar la experiencia del usuario, se proyecta desarrollar una Progressive Web App (PWA), la cual combina lo mejor de las aplicaciones web y móviles [8]. Posibilita su instalación en la pantalla de inicio de dispositivos móviles, proporcionando un acceso rápido y una experiencia de usuario inmersiva. Las PWAs son responsivas, adaptándose a diferentes tamaños de pantalla y dispositivos, ofreciendo una experiencia rápida y confiable, con una interfaz similar a la de una aplicación nativa. Esta aplicación resulta una solución práctica para mostrar los datos de los cursos, resultando ser conveniente tanto para el desarrollo como para el mantenimiento al utilizar tecnologías web estándar.

También se utiliza la tecnología Docker [9] a nivel de infraestructura, dando soporte al entorno web, simplificando el despliegue y la gestión de la aplicación, y permitiendo por consiguiente la escalabilidad de la misma.

Para la extracción de los datos, se desarrolla un bot para hacer Web Scraping con Python. La selección de Python como lenguaje de programación se debe a varias razones: en primer lugar, la existencia de bibliotecas como BeautifulSoup y Scrapy que proporcionan funciones básicas para la extracción de datos; y por otro lado, constituye un lenguaje sencillo de programar.

Para este proyecto en particular, se utiliza la biblioteca BeautifulSoup para analizar documentos HTML [10]. Esta herramienta transforma un documento HTML en un árbol de objetos de Python, permitiendo navegar y extraer datos de manera eficiente. De esta manera, BeautifulSoup se emplea específicamente para extraer la información de los cursos ofrecidos por las diferentes Facultades Regionales de la Universidad Tecnológica Nacional.

3 Resultados parciales

Hasta el momento, se avanzó significativamente en el proyecto. La interfaz del sitio web fue definida y codificada, y se diseñó la estructura de la API, que actúa como intermediaria entre los datos obtenidos mediante web scraping, alojados en un archivo json, y la interfaz gráfica(Ver Fig. 1).

La extracción de datos de los sitios web de las facultades de la Universidad Tecnológica Nacional (UTN) fue completada con éxito en dos regionales, incluyendo la recopilación de información sobre la oferta de cursos disponible en las plataformas de E-Learning.

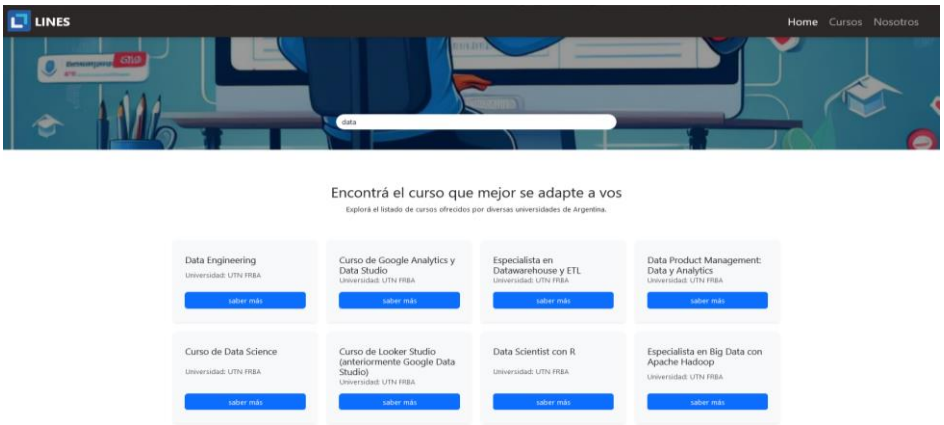

Fig. 1. Vista de interfaz de usuario.

4 Conclusiones. Líneas de Investigación Futuras

La implementación exitosa de la interfaz de usuario y la estructura de la API, junto con la extracción inicial de datos de las dos facultades regionales de la UTN, demostró el potencial del proyecto para transformar la manera en que se accede a la oferta de cursos de extensión. Estos logros iniciales establecieron una base sólida para el desarrollo continuo de la aplicación.

A medida que el proyecto avanza, el proceso de web scraping, esencial para la recopilación de datos, seguirá siendo iterativo. La integración de datos adicionales de otras facultades regionales de la UTN y, en una fase posterior, de otras Universidades Públicas Argentinas, permitirá enriquecer la base de datos y ampliar la cobertura de la aplicación.

En este sentido, se vislumbra que con el crecimiento de la cantidad de información disponible será necesario implementar opciones avanzadas de filtrado para facilitar a los usuarios la búsqueda de cursos específicos. Esta funcionalidad mejorará la experiencia del usuario al permitir una navegación más eficiente y personalizada dentro de la aplicación.

En resumen, mientras se consolida la infraestructura básica del proyecto, el enfoque estará en la expansión de la base de datos, la optimización de la experiencia del usuario y la exploración de nuevas tecnologías. Estos esfuerzos maximizarán el impacto de la aplicación en la comunidad educativa y contribuirán significativamente a la mejora del acceso a oportunidades de educación continua en Argentina.

Bibliografia

1. Martínez, R., Rodríguez, R. A., Vera, P., Parkinson, C.: Análisis de técnicas de raspado de datos en la web aplicado al Portal del Estado Nacional Argentino. In XXV Congreso Argentino de Ciencias de la Computación (CACIC) (Universidad Nacional de Río Cuarto, Córdoba, 14 al 18 de octubre de 2019).
2. Mitchell, R.: Web scraping with Python: Collecting more data from the modern web. " O'Reilly Media, Inc." (2018).
3. De Sanctis, V.: Building Web APIs with ASP.NET Core. Apress (2023).
4. Johnson, C., Elliott, R.: Data Extraction Techniques. New York: Academic Press (2006).
5. Russell, M. A.: Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Instagram, GitHub, and More. O'Reilly Media (2019).
6. Prieto Roig, A.: Web Data Scraper (Doctoral dissertation, Universitat Politècnica de València) (2023).
7. Bootstrap. (n.d.). Bootstrap Documentation. https://getbootstrap.com/docs/, last accessed 2024/07/24.
8. Firtmann, M.: Progressive Web Apps in 2020. The Medium. (2020): https://medium.com/@firt/progressive-web-apps-in-2020-c15018c9931c, last accessed 2024/07/12.
9. Docker, Inc. (n.d.). Docker Documentation. Retrieved from https://docs.docker.com/
10. Richardson, L.: Beautiful Soup Documentation (2020). https://www.crummy.com/software/BeautifulSoup/bs4/doc/, last accessed 2024/07/07.