Observatorio de demanda laboral en América Latina

Nicolas Francisco Camacho Alarcón
Alejandro Pinzón Fajardo

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
SYSTEMS ENGINEERING PROGRAM
BOGOTÁ, D.C.
2025

¡CODE¿

Observatorio de demanda laboral en América Latina

Author(s):

Nicolas Francisco Camacho Alarcón
Alejandro Pinzón Fajardo

UNDERGRADUATE FINAL PROJECT REPORT PERFORMED IN ORDER TO ACCOMPLISH ONE OF THE REQUIREMENTS FOR THE SYSTEMS ENGINEERING DEGREE

Director

Ing. Luis Gabriel Moreno Sandoval

Juries of the Undergraduate Final Project

Ing. ¡Name of the jury¿

Ing. ¡Name of the jury¿

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERIA

SYSTEMS ENGINEERING PROGRAM

BOGOTÁ, D.C.

¡Month¿, 2025

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
SYSTEMS ENGINEERING PROGRAM

President of the Pontificia Universidad Javeriana

¡Name of the President of the University¿

Dean of School of Engineering

¡Name of the Dean¿

Head of the Systems Engineering Program

¡Name of the head of the program¿

Head of the Systems Engineering Department

¡Name of the head of the department¿

Artículo 23 de la Resolución No. 1 de Junio de 1946

"La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia"

GRATITUDE

Write a message if you feel gratitude for someone who has supported the development of the project. Your family, your partner, your friends, your principal, teachers, etc.

CONTENT

ABSTRACT

El desajuste entre las habilidades demandadas por el mercado y la oferta formativa en Latinoamérica dificultaba decisiones de política, academia y empresa. Este proyecto abordó el problema construyendo un observatorio automatizado que recolectó avisos de empleo multi-portal y multi-país, escalable hacia $\sim$600.000 registros. Se integraron spiders (Scrapy/Selenium con anti-detección), una base PostgreSQL con pgvector y un pipeline de extracción/-normalización de habilidades (NER/regex con apoyo LLM) alineadas a ESCO. El sistema generó indicadores, consultas y visualizaciones reproducibles, entregando evidencia comparable por país, sector y tiempo para orientar currículos, formación y estrategias de talento.

Capítulo 1

INTRODUCCIÓN

Los mercados laborales latinoamericanos evolucionan con rapidez y publican sus vacantes en portales heterogéneos, con formatos dispares, vocabularios no estandarizados y alta volatilidad (los avisos desaparecen o cambian con frecuencia). Esta fragmentación dificulta medir, con evidencia objetiva y comparable, qué habilidades técnicas y digitales están siendo demandadas por país, sector y momento del tiempo. El proyecto **Observatorio de Demanda Laboral para América Latina** responde a ese vacío mediante un sistema automatizado que captura, estructura y analiza anuncios de empleo a escala.

La solución integra un **pipeline** modular de ocho etapas: Scraping multifuente y multipaís (Colombia, México, Argentina) con spiders robustos y medidas anti-detección; Normalización y limpieza; Extracción de habilidades combinando NER, patrones regex y apoyo LLM; Alineación a la taxonomía ESCO; Generación de embeddings multilingües (modelo E5); Reducción dimensional (UMAP); Clustering (HDBSCAN) para descubrir familias de perfiles; y Visualización y reportes. La infraestructura técnica se apoya en **Python/Scrapy, PostgreSQL + pgvector** y **Docker**, con registro y monitorización de extremo a extremo.

Este documento guía al lector desde el contexto y la motivación hasta los resultados y conclusiones. Presenta: I) antecedentes y trabajos relacionados; II) arquitectura del sistema y orquestación; III) adquisición y modelado de datos; IV) métodos de extracción y normalización de habilidades; V) componentes de representación y análisis; VI) evaluación y métricas; VII) hallazgos y visualizaciones; VIII) consideraciones éticas y limitaciones; y IX) conclusiones y trabajo futuro.

Capítulo 2

CONTEXTO DEL PROBLEMA

2.1 Oportunidad y problema

2.1.1 Contexto del problema

El mercado laboral en América Latina se encontró, durante la última década, en una compleja encrucijada definida por la confluencia de dos fuerzas a menudo contrapuestas: una acelerada transformación digital y la persistencia de desafíos estructurales, como una elevada informalidad laboral y brechas de capital humano (Echeverría & Rucci, 2022). La pandemia de COVID-19 actuó como un catalizador sin precedentes, intensificando la adopción de tecnologías y, con ello, la demanda de competencias digitales, al tiempo que exponía la vulnerabilidad de los mercados de trabajo de la región (Azuara et al., 2022). Este dinamismo generó el riesgo de que la automatización y la digitalización, de no ser gestionadas estratégicamente, pudiesen exacerbar las desigualdades existentes, conduciendo a una mayor polarización y segmentación social (Echeverría & Rucci, 2022).

Para analizar este fenómeno regional de manera tangible y robusta, este proyecto seleccionó como casos de estudio a tres de las economías más grandes y digitalmente activas de habla hispana: Colombia, México y Argentina. La elección de estos países respondió a tres criterios estratégicos. Primero, su alto volumen de publicaciones de ofertas laborales en portales digitales aseguró la viabilidad de una recolección masiva de datos (web scraping), fundamental para el entrenamiento de modelos de lenguaje robustos (**rubio2024**; Aguilera & Méndez, 2018; Martínez Sánchez, 2024). Segundo, la existencia de estudios previos en cada país, aunque metodológicamente limitados, confirmó la pertinencia del problema y proporcionó una línea de base para la comparación (Cárdenas Rubio et al., 2015). Y tercero, su

diversidad en términos de realidades económicas, territoriales y de madurez digital permitió validar que la solución desarrollada fuese portable y adaptable a los distintos contextos que caracterizan a América Latina.

2.1.2 El caso de Colombia: cambio estructural en la demanda de habilidades

El caso de Colombia sirvió como una ilustración profunda de esta dinámica. El diagnóstico nacional previo al proyecto ya indicaba que el principal cuello de botella para la inclusión digital no era la falta de infraestructura, sino la brecha de capital humano. Específicamente, el "Índice de Brecha Digital" (IBD) del Ministerio de Tecnologías de la Información y las Comunicaciones reveló que la dimensión de "Habilidades Digitales" constituía el mayor componente individual de la brecha en el país.

Esta evidencia fue posteriormente corroborada y cuantificada por el análisis empírico de la demanda laboral, el cual demostró que la pandemia generó un cambio estructural y persistente en el mercado. Se encontró que, en los 18 meses posteriores al inicio de la crisis sanitaria, las vacantes tecnológicas aumentaron en un 50 % en comparación con las no tecnológicas (**rubio2024**).

Este cambio no fue solo cuantitativo, sino también cualitativo: se observó una marcada caída en la demanda de herramientas ofimáticas tradicionales como Excel (cuya mención en ofertas cayó del 35.8 % en 2018 al 17.4 % en 2023) y un surgimiento exponencial de tecnologías especializadas asociadas al desarrollo web y la gestión de datos, como bases de datos NoSQL (12.3 %), el framework Django (5.5 %) y la librería React (5.3 %) para el año 2023 (**rubio2024**).

2.1.3 Realidades en México y Argentina

De manera similar, en México se ha documentado un desajuste significativo entre la oferta educativa y la demanda laboral en el sector tecnológico. El análisis de miles de ofertas laborales reveló que las empresas demandan perfiles con habilidades especializadas en desarrollo de software, análisis de datos y ciberseguridad, mientras que la oferta de graduados se concentra en áreas tradicionales de sistemas (Martínez Sánchez, 2024).

En Argentina, el sector de Tecnologías de la Información ha experimentado un crecimiento sostenido, convirtiéndose en uno de los principales generadores de empleo calificado. Sin embargo, la identificación de las tecnologías y roles más demandados ha sido un desafío

constante para instituciones educativas y agencias de empleo (Aguilera & Méndez, 2018).

2.1.4 Formulación del problema

A pesar de que el contexto del problema —la creciente e insatisfecha demanda de habilidades tecnológicas— estaba claramente identificado, los métodos existentes en la región para analizarlo presentaban limitaciones metodológicas significativas que impedían una comprensión profunda y ágil del fenómeno.

Los estudios de referencia en los países seleccionados, si bien valiosos para establecer tendencias macro, se basaron en enfoques de análisis léxico y reglas manuales. En Colombia, el análisis se centró en un sistema de clasificación basado en la Clasificación Internacional Uniforme de Ocupaciones (CIUO), utilizando algoritmos de emparejamiento de texto con tokenización y métricas de similitud basadas en n-gramas (**rubio2024**). De forma análoga, en Argentina, los estudios se concentraron en técnicas de minería de texto con análisis de frecuencias y bigramas para identificar patrones en las ofertas del sector TI (Aguilera & Méndez, 2018). En México, el enfoque combinó datos de encuestas con scraping de portales, apoyándose en el análisis de frecuencia de términos y la creación de tipologías manuales para segmentar las habilidades (Martínez Sánchez, 2024).

La limitación fundamental compartida por estos enfoques es su dependencia de la correspondencia léxica explícita, lo que los hace incapaces de capturar la riqueza semántica del lenguaje. Estos métodos no podían:

- Detectar habilidades implícitas (aquellas que se infieren del contexto de un cargo pero no se mencionan directamente)
- Gestionar la ambigüedad del lenguaje informal o el uso de anglicismos técnicos (“Spanglish”)
- Identificar clústeres de competencias emergentes que aún no forman parte de taxonomías estandarizadas

La alta variabilidad en la redacción de las ofertas laborales, la falta de estructuras normalizadas y la rápida aparición de nuevas tecnologías hacían que estos sistemas fueran metodológicamente frágiles y requirieran un constante mantenimiento manual (Echeverría & Rucci, 2022).

2.1.5 Vacío identificado a nivel institucional

Más allá de las limitaciones académicas individuales, esta carencia de infraestructura analítica ha sido reconocida a nivel institucional. El Banco Interamericano de Desarrollo (BID) ha señalado la falta de pipelines de análisis modernos y automatizados en la región, destacando que la mayoría de los sistemas existentes, si bien articulan el scraping, todavía se basan en reglas fijas o mapeos manuales y no han incorporado técnicas de embeddings ni de NLP avanzado (Echeverría & Rucci, 2022).

En consecuencia, el problema específico que este proyecto abordó fue la ausencia de una herramienta automatizada y de extremo a extremo que, adaptada a las particularidades lingüísticas y estructurales del español latinoamericano, permitiera superar las limitaciones de los análisis léxicos tradicionales. Se identificó la necesidad de un sistema capaz de extraer, estructurar y analizar la evolución de las habilidades tecnológicas de manera semántica, escalable y con un mayor grado de autonomía, integrando para ello técnicas avanzadas de Procesamiento de Lenguaje Natural (NLP), enriquecimiento contextual con Large Language Models (LLMs) y algoritmos de agrupamiento no supervisado.

2.2 Propuesta de solución

Para dar respuesta al problema formulado, se diseñó e implementó un observatorio de demanda laboral tecnológica basado en un pipeline modular y automatizado, un proyecto enmarcado en las áreas de Ingeniería de Sistemas y Ciencia de Datos. El sistema fue concebido como una solución de extremo a extremo que integró las etapas de recolección, procesamiento, análisis semántico y segmentación de ofertas de empleo publicadas en Colombia, México y Argentina. El objetivo fue crear una arquitectura robusta, replicable y adaptada a las complejidades del contexto latinoamericano, superando las limitaciones de los enfoques puramente léxicos o manuales.

La solución se materializó a través de un sistema compuesto por módulos secuenciales y cohesivos. El primer módulo consistió en un motor de adquisición de datos que, mediante técnicas de web scraping, extrajo de forma sistemática y ética decenas de miles de ofertas laborales de portales de empleo clave en la región. El núcleo del sistema fue su arquitectura de extracción dual, compuesta por dos pipelines paralelos:

Pipeline A (Tradicional): Implementó un método de extracción basado en Reconocimiento de Entidades Nombradas (NER) utilizando un EntityRuler de spaCy, poblado con la taxonomía completa de ESCO, combinado con expresiones regulares para capturar un base-

line de habilidades explícitas de alta precisión.

Pipeline B (Basado en LLMs): Empleó Large Language Models (LLMs) como Llama 3 para realizar una extracción semántica, capaz de identificar no solo habilidades explícitas sino también de inferir competencias implícitas a partir del contexto de la vacante.

Posteriormente, un módulo de mapeo de dos capas normalizó las habilidades extraídas por ambos pipelines contra la taxonomía ESCO. La primera capa realizó una coincidencia léxica (exacta y difusa), mientras que la segunda ejecutó una búsqueda de similitud semántica de alto rendimiento, utilizando embeddings multilingües (E5) y un índice FAISS pre-calculado. Finalmente, un módulo de análisis no supervisado aplicó una secuencia metodológica de embeddings, reducción de dimensionalidad con UMAP y agrupamiento con HDBSCAN para identificar clústeres de habilidades y perfiles emergentes.

2.3 Justificación de la solución

La solución implementada se justificó como una alternativa superior y mejor adaptada para el análisis de la demanda de habilidades en América Latina, ya que abordó directamente las debilidades metodológicas identificadas en los estudios previos. A diferencia de los enfoques basados exclusivamente en reglas léxicas (**rubio2024**; Aguilera & Méndez, 2018), la arquitectura de dos pipelines paralelos permitió una validación empírica cruzada: combinó la auditabilidad y alta precisión para habilidades conocidas del Pipeline A con la potencia inferencial y la capacidad de descubrir habilidades implícitas del Pipeline B.

Este diseño comparativo proveyó un marco para evaluar objetivamente el rendimiento de los LLMs, en lugar de depender únicamente de su capacidad "black-box".

Técnicamente, el sistema representó un avance significativo en escalabilidad y eficiencia. La implementación de un índice FAISS para la búsqueda semántica de similitud permitió procesar grandes volúmenes de datos a una velocidad órdenes de magnitud superior a las búsquedas en bases de datos vectoriales convencionales, haciendo factible el análisis de todo el corpus recolectado.

Adicionalmente, el sistema fue diseñado explícitamente para la realidad del español latinoamericano. Este enfoque abordó directamente una limitación crítica de trabajos de vanguardia en LLMs, los cuales se han desarrollado y validado casi exclusivamente sobre datasets en inglés, ignorando las particularidades lingüísticas (como el "Spanglish") del dominio tecnológico en la región.

Finalmente, el valor agregado del proyecto residió en su síntesis estratégica de metodo-

logías de vanguardia. El sistema no se limitó a una sola técnica, sino que articuló la cobertura del scraping regional, la potencia de los LLMs ajustados para generar salidas estructuradas, y la capacidad estructuradora del clustering semántico. Al hacerlo, se desarrolló un observatorio más completo, robusto y metodológicamente transparente que las alternativas existentes, estableciendo una base sólida y replicable para el monitoreo dinámico de la demanda laboral en la región.

2.4 Descripción del proyecto

El proyecto se concibió como un observatorio automatizado para capturar, normalizar y analizar avisos de empleo en Latinoamérica. Se integraron múltiples portales (CO, MX y AR), se diseñó una base de datos relacional con soporte vectorial, y se implementó un pipeline de extracción de habilidades (NER/regex/LLM) alineadas a ESCO, con generación de indicadores, visualizaciones y reportes. Operativamente, se planificó escalar hasta 600.000 avisos para la defensa, garantizando calidad, trazabilidad y reproducibilidad.

2.4.1 Objetivo general

Desarrollar un sistema que permita procesar y segmentar la demanda de habilidades tecnológicas en Colombia, México y Argentina, mediante técnicas de procesamiento de lenguaje natural.

2.4.2 Objetivos específicos

- Construir un estado del arte exhaustivo para comparar trabajos existentes en el ámbito de observatorios laborales automatizados y técnicas de procesamiento de lenguaje natural en español.
- Diseñar una arquitectura modular, escalable y reutilizable para el observatorio laboral automatizado, fundamentada en las mejores prácticas identificadas en el estado del arte.
- Implementar e integrar técnicas de inteligencia artificial para la identificación, normalización y agrupación semántica de habilidades tecnológicas en ofertas laborales en español.
- Validar el desempeño y la robustez de la arquitectura y los modelos propuestos mediante métricas cuantitativas y estudios empíricos.

2.4.3 Entregables, estándares y justificación

Entregable	**Estándares asociados**	**Justificación**
Repositorio de código (spiders, orquestador, pipelines)	PEP 8/257/484; Conv. Commits; SemVer	Mantenibilidad, legibilidad y control de versiones.
Esquema BD y migraciones (PostgreSQL + pgvector)	Normalización (3NF); SQL best practices	Integridad, trazabilidad y soporte a consultas vectoriales.
Spiders y configuración de scraping	Polite crawling (delays/retries); manejo anti-bots	Captura estable a escala y resiliencia ante cambios UI.
Orquestador CLI + scheduler	CLI UX (Typer); jobs idempotentes	Operación reproducible, programable y auditable.
Módulo de extracción/normalización de habilidades	ISO/IEC/IEEE 29148 (requisitos); ESCO	Consistencia semántica y comparabilidad entre países.
Embeddings y análisis (E5, UMAP, HDBSCAN)	Procedimientos reproducibles; semillas fijas	Descubrimiento de patrones y replicabilidad experimental.
Datasets consolidados (CSV/JSON) + diccionario de datos	Esquemas declarativos; control de versiones	Consumo externo y verificación de calidad.
Documentación técnica y de proyecto (SRS, SPMP, VFP, manuales)	IEEE 1058 (plan de proyecto); 29148 (requisitos)	Alineación con buenas prácticas y transferencia de conocimiento.
Reportes y visualizaciones (PDF/PNG/CSV)	Principios de visualización; metadatos	Comunicación clara de hallazgos a públicos no técnicos.
Plan de operación y mantenimiento (Docker/monitoring)	Buenas prácticas Docker/-Logging	Despliegue consistente y observabilidad del sistema.

Capítulo 3

MARCO TEÓRICO

Para comprender el diseño y la justificación de la solución desarrollada, es necesario fundamentar el proyecto en una serie de conceptos clave provenientes de la ingeniería de sistemas, la ciencia de datos y, fundamentalmente, del Procesamiento de Lenguaje Natural (NLP). Estos conceptos no actúan de forma aislada, sino que se articulan en un flujo metodológico que va desde la adquisición de datos brutos hasta la generación de conocimiento estructurado sobre el mercado laboral.

3.1 Web Scraping y Adquisición de Datos

El punto de partida del observatorio es la recolección de datos a gran escala desde fuentes web públicas. Esta tarea se realiza mediante **Web Scraping**, una técnica de extracción automatizada de información desde el código HTML de las páginas web (Orozco Puello & Gómez Estrada, 2019). En el contexto del mercado laboral, esta técnica ha demostrado ser fundamental para obtener datos de alta frecuencia y granularidad directamente de los portales de empleo, superando las limitaciones de las encuestas y los reportes institucionales, que suelen ser retrospectivos y de baja periodicidad (**rubio2024**; Cárdenas Rubio et al., 2015).

3.1.1 Conceptos clave del web scraping

El web scraping se distingue del simple *crawling* en que no solo navega páginas web, sino que extrae y estructura información específica. Las técnicas modernas incluyen:

- **Parsers HTML**: Herramientas como BeautifulSoup y lxml que permiten navegar y extraer elementos del Document Object Model (DOM).

- **Headless browsers**: Tecnologías como Playwright y Puppeteer que permiten ejecutar JavaScript y capturar contenido dinámico.
- **Control de rate limiting**: Técnicas de throttling y backoff exponencial para respetar los límites de los servidores.
- **Rotación de user-agents y proxies**: Estrategias para distribuir las peticiones y evitar bloqueos.

3.1.2 Buenas prácticas y consideraciones éticas

La implementación de web scraping debe seguir principios éticos y legales, incluyendo:

- Respeto del archivo `robots.txt`
- Implementación de delays entre peticiones
- Registro de fuentes y sellos de tiempo
- Validación y normalización de datos extraídos
- Monitoreo de cambios en la estructura del DOM

3.2 Procesamiento de Lenguaje Natural (NLP)

Una vez extraído el contenido textual de las ofertas laborales, el siguiente paso es prepararlo para el análisis computacional mediante técnicas de Procesamiento de Lenguaje Natural.

3.2.1 Preprocesamiento de texto

El preprocesamiento es fundamental para estandarizar y limpiar los datos textuales:

- **Tokenización**: Segmentación del texto en unidades mínimas o "tokens" (generalmente palabras o signos de puntuación) (Nguyen et al., 2024). Este proceso transforma cadenas de texto continuo en secuencias discretas procesables computacionalmente.
- **Lematización**: Reducción de las palabras a su forma base o raíz gramatical (lema), permitiendo agrupar variaciones morfológicas de un mismo término. Por ejemplo, "programar", "programando" y "programado" se unifican bajo el lema "programar" (Echeverría & Rucci, 2022). Este paso es crucial para estandarizar el vocabulario y reducir la dispersión de los datos antes del análisis.

3.2.2 Extracción de habilidades con NER y Regex

Con el texto limpio y normalizado, el núcleo del desafío consiste en la extracción de habilidades. Para ello, se emplea un enfoque híbrido:

- **Expresiones Regulares (Regex)**: Un lenguaje de patrones sintácticos que permite identificar y extraer secuencias de texto muy específicas, como nombres de tecnologías o certificaciones con formatos predecibles (Lukauskas et al., 2023). Son especialmente efectivas para capturar menciones explícitas de tecnologías con nomenclaturas estandarizadas.
- **Reconocimiento de Entidades Nombradas (NER)**: Una técnica de NLP diseñada para identificar y clasificar entidades en un texto, como nombres de personas, lugares o, en este caso, habilidades y competencias (Herandi et al., 2024). El NER permite pasar de una búsqueda basada en reglas a un sistema capaz de reconocer habilidades en contextos gramaticales complejos.

3.3 Large Language Models (LLMs)

Para superar las limitaciones de la extracción de menciones explícitas, el proyecto incorpora el uso de **Large Language Models (LLMs)**. Estos modelos de lenguaje a gran escala, como GPT o Llama 3, entrenados sobre corpus masivos de texto, poseen capacidades de razonamiento contextual que permiten abordar desafíos más complejos (Herandi et al., 2024).

3.3.1 Prompt Engineering

A través del diseño de prompts específicos (**Prompt Engineering**), es posible guiar a los LLMs para realizar tareas de enriquecimiento semántico, como:

- Distinción entre habilidades explícitas (mencionadas textualmente) e implícitas (inferidas del contexto del cargo) (Nguyen et al., 2024)
- Normalización de variantes terminológicas
- Clasificación de habilidades según taxonomías predefinidas
- Generación de salidas estructuradas en formatos como JSON

3.3.2 Estrategias de uso de LLMs

Existen diferentes modalidades de aplicación de LLMs para extracción de habilidades:

- **Zero-shot learning**: El modelo realiza la tarea sin ejemplos previos, basándose únicamente en la instrucción del prompt.
- **Few-shot learning**: Se proporcionan algunos ejemplos en el prompt para guiar el comportamiento del modelo (Nguyen et al., 2024).
- **Fine-tuning**: Re-entrenamiento del modelo sobre datasets específicos del dominio para mejorar su rendimiento (Herandi et al., 2024; Zhang et al., 2022).

3.4 Embeddings Semánticos y Representación Vectorial

Una vez extraídas y normalizadas, las habilidades deben ser representadas de una forma que permita su análisis cuantitativo. Para ello, se utilizan **Embeddings Semánticos**, que son representaciones vectoriales (numéricas) de palabras o frases en un espacio de alta dimensionalidad (Kavas et al., 2024).

3.4.1 Propiedades de los embeddings

La propiedad fundamental de estos embeddings es que la distancia entre dos vectores en ese espacio refleja la similitud semántica entre los textos que representan. Esto permite:

- Capturar relaciones semánticas complejas
- Realizar búsquedas por similitud de manera eficiente
- Agrupar habilidades relacionadas
- Comparar ofertas laborales en un espacio vectorial continuo

3.4.2 Embeddings Multilingües

Dado que las ofertas laborales en América Latina a menudo contienen términos técnicos en inglés (“Spanglish”), es crucial el uso de **Embeddings Multilingües**, modelos entrenados para que textos con el mismo significado en diferentes idiomas tengan representaciones

vectoriales cercanas en el mismo espacio semántico (Echeverría & Rucci, 2022; Kavas et al., 2024).

Modelos populares incluyen:

- **E5-large**: Modelo de embeddings multilingüe de alta calidad
- **Sentence Transformers**: Familia de modelos basados en BERT optimizados para generar embeddings de oraciones
- **multilingual-e5-base**: Versión multilingüe del modelo E5 base

3.5 Análisis No Supervisado: Reducción de Dimensionalidad y Clustering

Finalmente, para descubrir patrones y estructuras latentes en el conjunto de datos, se aplica un pipeline de análisis no supervisado.

3.5.1 Reducción de Dimensionalidad con UMAP

Debido a que los embeddings son vectores de muy alta dimensionalidad (ej. 768 dimensiones), lo que dificulta la efectividad de muchos algoritmos (la "maldición de la dimensionalidad"), primero se aplica una técnica de **Reducción de Dimensionalidad** como **UMAP (Uniform Manifold Approximation and Projection)**.

UMAP es un algoritmo no lineal que reduce el número de dimensiones preservando tanto la estructura local como global de los datos, lo que lo hace superior a métodos lineales como PCA para visualizar relaciones semánticas complejas (Lukauskas et al., 2023).

3.5.2 Clustering con HDBSCAN

Sobre los datos ya en un espacio de baja dimensionalidad, se aplica un algoritmo de **Clustering** basado en densidad como **HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise)**.

A diferencia de métodos como K-Means, HDBSCAN posee las siguientes ventajas (Lukauskas et al., 2023):

- No requiere especificar el número de clústeres de antemano

- Es capaz de identificar grupos de formas arbitrarias
- Puede separar los puntos que no pertenecen a ningún grupo como “ruido”
- Funciona bien con clústeres de densidades variables

Esta secuencia metodológica, inspirada en la literatura de vanguardia, es la que permite la identificación automática de “ecosistemas de habilidades” y perfiles laborales emergentes (Lukauskas et al., 2023).

3.6 Taxonomías Estandarizadas: ESCO e ISCO

Para asegurar la comparabilidad y estandarización de los resultados, el sistema integra taxonomías internacionales reconocidas:

3.6.1 ESCO (European Skills, Competences, Qualifications and Occupations)

ESCO es una taxonomía multilingüe desarrollada por la Comisión Europea que clasifica y organiza habilidades, competencias, calificaciones y ocupaciones relevantes para el mercado laboral de la UE (Kavargyris et al., 2025). Contiene más de 13,000 habilidades y conocimientos catalogados de manera jerárquica.

3.6.2 ISCO-08 (International Standard Classification of Occupations)

ISCO es un estándar internacional mantenido por la Organización Internacional del Trabajo (OIT) para clasificar ocupaciones. Proporciona un marco común para comparar datos ocupacionales entre países.

3.6.3 Integración con O*NET

Para el contexto de habilidades tecnológicas generales, también se toma como referencia **O*NET**, un portal de empleo financiado por el Departamento de Trabajo de EE. UU. que reporta habilidades de alta demanda en el mercado laboral estadounidense (**rubio2024**).

Capítulo 4

ESTADO DEL ARTE

El desafío de extraer, analizar y comprender la demanda de habilidades a partir de fuentes de datos no estructuradas, como las ofertas de empleo en línea, ha sido abordado desde múltiples frentes en la literatura académica y aplicada. Si bien el objetivo es común —traducir texto en conocimiento accionable sobre el mercado laboral—, las aproximaciones metodológicas varían significativamente en su complejidad, escalabilidad y profundidad semántica.

Para posicionar adecuadamente la contribución de este proyecto, fue necesario realizar un análisis crítico de las soluciones existentes a nivel global, las cuales se pueden agrupar en tres grandes líneas de trabajo:

- **Enfoques Regionales en América Latina**: Un conjunto de estudios pioneros en la región que validaron el uso de técnicas de web scraping para la recolección de datos, pero cuyo análisis se fundamentó en métodos de Procesamiento de Lenguaje Natural (NLP) tradicionales, como el análisis léxico y el emparejamiento basado en reglas.

- **La Frontera de la Extracción con Large Language Models (LLMs)**: Investigaciones de vanguardia a nivel internacional que exploraron el uso de modelos de lenguaje a gran escala, tanto en modalidades de prompting (sin re-entrenamiento) como de fine-tuning (con re-entrenamiento), para lograr una extracción de habilidades con mayor capacidad semántica e inferencial.

- **Pipelines Semánticos y Descubrimiento No Supervisado**: Arquitecturas de análisis completas que, más allá de la extracción, integran embeddings semánticos, técnicas de reducción de dimensionalidad y algoritmos de clustering para descubrir patrones y perfiles laborales emergentes directamente desde los datos.

El siguiente análisis demostrará que, si bien cada una de estas líneas ha aportado herramientas y hallazgos valiosos, ninguna de ellas, de forma aislada, resolvía de manera integral los desafíos metodológicos, geográficos y lingüísticos que presenta el mercado laboral tecnológico en América Latina. Esta fragmentación en el estado del arte fue la que justificó la necesidad de una solución sintética y adaptada, como la que se desarrolló en este proyecto.

4.1 Enfoques Regionales: Caracterización del Mercado con Métodos Léxicos

La primera línea de trabajo relevante para este proyecto comprende un conjunto de estudios pioneros desarrollados en América Latina. Estos trabajos fueron fundamentales porque validaron el uso de portales de empleo en línea como una fuente de datos rica y de alta frecuencia para el análisis del mercado laboral, pero se caracterizaron por emplear metodologías de procesamiento de texto basadas en análisis léxico, frecuencias de términos y reglas manuales.

4.1.1 El caso colombiano: análisis de la transformación digital

El estudio más completo y reciente en este ámbito fue el de Rubio Arrubla (2025) para el mercado colombiano. Este trabajo construyó una base de datos masiva mediante web scraping del portal elempleo.com para el periodo 2018-2023, abarcando más de 500,000 ofertas laborales. Su principal aporte fue la caracterización cuantitativa del impacto de la pandemia, demostrando un cambio estructural en la demanda de habilidades.

Metodológicamente, el estudio implementó una tipología propia de habilidades tecnológicas dividida en cuatro categorías: (1) Teletrabajo, (2) Habilidades tecnológicas generales, (3) Habilidades tecnológicas especializadas, y (4) Habilidades de TIC. El proceso de clasificación de las vacantes utilizó un algoritmo de emparejamiento de texto basado en la descomposición de textos en n-gramas y el cálculo de puntajes de similitud contra la Clasificación Internacional Uniforme de Ocupaciones (CIUO) (Rubio Arrubla, 2025).

Los hallazgos cuantitativos fueron reveladores: en los 18 meses posteriores al inicio de la crisis sanitaria, las vacantes tecnológicas aumentaron en un 50 % en comparación con las no tecnológicas. Este cambio no fue solo cuantitativo, sino también cualitativo: se observó una marcada caída en la demanda de herramientas ofimáticas tradicionales como Excel (cuya mención en ofertas cayó del 35.8 % en 2018 al 17.4 % en 2023) y un surgimiento exponencial

de tecnologías especializadas asociadas al desarrollo web y la gestión de datos, como bases de datos NoSQL (12.3 %), el framework Django (5.5 %) y la librería React (5.3 %) para el año 2023 (Rubio Arrubla, 2025).

Sin embargo, la dependencia de la coincidencia léxica representó una limitación fundamental. El propio estudio reconoció que el método perdía eficiencia a medida que aumentaba el número de palabras en los títulos, al no poder capturar el contexto general (Rubio Arrubla, 2025). Esta debilidad ilustra el problema central de los enfoques basados exclusivamente en n-gramas: la incapacidad de procesar la riqueza semántica del lenguaje natural.

4.1.2 Experiencias en Argentina y México

De forma análoga, el trabajo de Aguilera y Méndez (2018) para el contexto argentino se centró en el sector de Tecnologías de la Información (TI), extrayendo datos de portales como ZonaJobs y Bumeran. Su análisis se apoyó en técnicas de minería de texto, específicamente en el análisis de frecuencias y el uso de bigramas, para identificar las tecnologías y roles más demandados (Aguilera & Méndez, 2018).

Sin embargo, para estandarizar el vocabulario informal de las ofertas, los autores tuvieron que construir una lista de palabras clave de forma semi-manual, lo que limita la escalabilidad del sistema y su capacidad para adaptarse a la aparición de nuevas tecnologías no contempladas inicialmente (Aguilera & Méndez, 2018).

Para el caso de México, la investigación de Martínez Sánchez (2024) propuso un enfoque innovador al combinar datos de encuestas oficiales con información obtenida mediante scraping. Su análisis se basó en la frecuencia de términos y en una tipología manual para segmentar las habilidades, arrojando luz sobre el desajuste entre oferta y demanda, pero sin incluir un procesamiento avanzado y automatizado del lenguaje natural (Martínez Sánchez, 2024).

4.1.3 Limitaciones sistémicas de los enfoques léxicos

En conjunto, estos estudios regionales fueron cruciales para establecer la viabilidad de la recolección de datos, pero, desde una perspectiva metodológica, expusieron una brecha fundamental compartida: su dependencia de la correspondencia léxica explícita. Al basarse en frecuencias de palabras, n-gramas o listas de términos predefinidos, estos sistemas eran metodológicamente frágiles ante la ambigüedad y la variabilidad del lenguaje natural.

Más allá de las limitaciones académicas individuales, esta carencia de infraestructura

analítica ha sido reconocida a nivel institucional. El Banco Interamericano de Desarrollo (BID) ha señalado la falta de pipelines de análisis modernos y automatizados en la región, destacando que la mayoría de los sistemas existentes, si bien articulan el scraping, todavía se basan en reglas fijas o mapeos manuales y no han incorporado técnicas de embeddings ni de NLP avanzado (Echeverría & Rucci, 2022).

Esta constatación institucional refuerza la conclusión de que existía un vacío sistémico: la ausencia de una solución que superara los enfoques léxicos para proporcionar un análisis semántico, dinámico y escalable de la demanda de habilidades en América Latina.

4.2 La Frontera de la Extracción: El Uso de Large Language Models

Paralelamente a los enfoques regionales, una segunda línea de investigación a nivel internacional ha explorado el uso de Large Language Models (LLMs) para superar las limitaciones de los métodos léxicos. Estos trabajos representan la frontera del estado del arte en extracción semántica, mostrando tanto el potencial transformador de los modelos de lenguaje de gran escala como las complejidades prácticas de su aplicación en dominios especializados como el mercado laboral.

4.2.1 Exploración con prompting: zero-shot y few-shot learning

Una de las primeras aproximaciones en este campo fue la de Nguyen et al. (2024), quienes investigaron el uso de LLMs de propósito general, como GPT-3.5 y GPT-4, en una modalidad de prompting sin re-entrenamiento (few-shot learning). Su metodología consistió en proporcionar al modelo una instrucción y unos pocos ejemplos de extracción de habilidades dentro del propio prompt.

Los investigadores experimentaron con dos formatos de salida: uno de extracción directa, donde el modelo devolvía una lista de habilidades (“EXTRACTION-STYLE”), y otro de etiquetado, donde el modelo reescribía la oración original encerrando las habilidades entre etiquetas especiales (“NER-STYLE”) (Nguyen et al., 2024).

Sus hallazgos fueron reveladores: aunque los LLMs no lograron igualar la precisión (medida con el F1-score) de los modelos supervisados tradicionales, demostraron una capacidad superior para interpretar frases sintácticamente complejas o ambiguas, como aquellas donde múltiples habilidades están conectadas por conjunciones (Nguyen et al., 2024).

Sin embargo, el estudio también advirtió sobre las limitaciones inherentes a este enfoque, principalmente:

- **Inconsistencia en los formatos de salida**: Los modelos producían resultados con estructuras variables, dificultando el parsing automático.
- **Riesgo de "alucinaciones"**: El modelo generaba entidades que no correspondían a habilidades reales presentes en el texto.
- **Rendimiento cuantitativo inferior**: En términos de F1-score, los resultados oscilaron entre 17.8 % y 27.8 %, muy por debajo de los modelos supervisados (Nguyen et al., 2024).

4.2.2 Fine-tuning de LLMs: Skill-LLM y el estado del arte

Tomando estas limitaciones como punto de partida, el trabajo de Herandi et al. (2024) representó la siguiente evolución lógica: el fine-tuning o re-entrenamiento específico de un LLM para la tarea. En su investigación, tomaron el modelo LLaMA 3 8B y lo ajustaron utilizando el dataset de referencia SkillSpan (Zhang et al., 2022).

Su principal innovación fue el diseño de un formato de salida estructurado en JSON que no solo extraía la habilidad (`skill_span`), sino también el contexto textual que la rodeaba. Este enfoque les permitió alcanzar un rendimiento que superó el estado del arte (SOTA), logrando un F1-score total de 64.8 %, superior tanto a los modelos supervisados previos como a los LLMs utilizados mediante prompting (Herandi et al., 2024).

Más importante aún, su método garantizó la consistencia y la auditabilidad de los resultados, resolviendo uno de los mayores problemas prácticos de los LLMs en modalidad de prompting. El sistema propuesto combinó:

- Uso de LoRA (Low-Rank Adaptation) para fine-tuning eficiente
- Entrenamiento en 2 epochs con batch size de 4 y learning rate de 2e-4
- Evaluación con Span F1 (exact match de spans)
- Distinción explícita entre "skills" (aplicación de conocimientos) y "knowledge" (saberes adquiridos)

Los resultados mostraron que Skill-LLM logró:

- F1 en skills: 54.3 %
- F1 en knowledge: 74.2 %
- F1 total: 64.8 % (vs. 64.2 % de NNOSE, 62.6 % de ESCOXLM-R, 58.9 % de JobSpanBERT) (Herandi et al., 2024)

4.2.3 La base de datos fundamental: SKILLSPAN

El trabajo de Herandi et al. (2024) se sustentó en SKILLSPAN, el primer dataset público a nivel de span para extracción de habilidades hard y soft presentado por Zhang et al. (2022). Este corpus contiene 391 postings anotados, 14.5K oraciones, 232K tokens, y 12.5K spans de habilidades/conocimientos, extraídos de tres fuentes (BIG, HOUSE, TECH) entre 2012-2021.

El proceso de anotación duró 8 meses, utilizando la herramienta Doccano, con múltiples rondas de ajuste de guías y evaluación de consistencia mediante Fleiss' $\kappa = 0.70$–0.75 (acuerdo sustancial) (Zhang et al., 2022). La diferenciación explícita entre "skills" (aplicación de conocimientos, ej. "work independently") y "knowledge" (saberes adquiridos, ej. "Python", "supply chain") fue fundamental para establecer un estándar de anotación robusto.

Los hallazgos del estudio de Zhang et al. (2022) demostraron que:

- El pre-entrenamiento continuo en textos de vacantes laborales mejora consistentemente sobre modelos generales
- STL (Single-Task Learning) supera a MTL (Multi-Task Learning): entrenar skills y knowledge por separado supera al modelo conjunto
- Extraer skills es más difícil (más largas, ambiguas, semánticas); knowledge es más fácil (tokens concretos, nombres propios)
- JobSpanBERT alcanzó ≈56.6 F1 en skills y JobBERT ≈63.9 F1 en knowledge (Zhang et al., 2022)

4.2.4 Limitación crítica: la barrera del idioma

A pesar de su sofisticación técnica, estos estudios de vanguardia comparten una limitación crucial que fue central para la justificación de este proyecto: fueron desarrollados y validados casi exclusivamente en contextos anglosajones y sobre datasets en idioma inglés.

El trabajo de Herandi et al. (2024), por ejemplo, se fundamentó íntegramente en el dataset SkillSpan, que contiene únicamente ofertas de empleo en inglés. Esta dependencia del idioma inglés evidenció un claro vacío geográfico y lingüístico en la aplicación de técnicas de NLP avanzadas para el análisis del mercado laboral.

En conclusión, si bien los LLMs representan la tecnología de punta para la extracción de habilidades, su aplicación efectiva no es trivial. El prompting simple resulta insuficiente en términos de precisión y consistencia (Nguyen et al., 2024), y las metodologías de fine-tuning de alto rendimiento, aunque superiores, estaban limitadas por la barrera del idioma de los datos de entrenamiento disponibles (Herandi et al., 2024).

4.3 Pipelines Semánticos y Descubrimiento No Supervisado

La tercera línea de investigación relevante se centra en arquitecturas de análisis completas que van más allá de la simple extracción de entidades para estructurar los datos y descubrir patrones latentes de manera no supervisada. Estos sistemas se enfocan en responder preguntas sobre cómo se agrupan las habilidades y cómo evolucionan los perfiles laborales, en lugar de solo identificar menciones individuales.

4.3.1 El pipeline de referencia: UMAP + HDBSCAN

El trabajo de Lukauskas et al. (2023) es el pilar fundamental de esta aproximación. Su investigación en el mercado laboral de Lituania propuso y validó empíricamente un pipeline de extremo a extremo que se ha convertido en una referencia metodológica. El flujo comenzaba con la extracción de las secciones de "Requisitos" de las ofertas de empleo mediante expresiones regulares (Regex).

A continuación, el texto extraído era vectorizado utilizando un modelo basado en BERT (Sentence Transformers) para generar embeddings semánticos de 384 dimensiones. Conscientes de la "maldición de la dimensionalidad", los autores compararon cinco métodos de reducción de dimensionalidad:

- PCA (lineal)
- t-SNE (no lineal, fuerte en visualización local)
- UMAP (no lineal, escalable, preserva estructura local y global)
- Trimap e Isomap como alternativas

El análisis concluyó que UMAP ofrecía los mejores resultados al preservar la estructura local y global de los datos de manera más efectiva que alternativas como PCA o t-SNE, medido según la métrica de trustworthiness (Lukauskas et al., 2023).

Finalmente, sobre los datos ya reducidos, aplicaron y compararon una batería de algoritmos de clustering:

- K-means: eficiente pero limitado a clusters esféricos
- DBSCAN / HDBSCAN: robustos frente a ruido, capturan clusters de densidad variable
- BIRCH: escalable a grandes volúmenes
- Affinity Propagation: no requiere número de clusters, pero costoso
- Spectral Clustering: captura relaciones no lineales

HDBSCAN mostró mejor estabilidad y capacidad para identificar clústeres de formas y densidades variables y manejar el ruido de manera robusta (Lukauskas et al., 2023).

El gran aporte de este estudio fue, por tanto, proporcionar una validación empírica para la secuencia completa:

Regex → Embeddings BERT → UMAP → HDBSCAN

como una metodología de vanguardia para el descubrimiento automático y no supervisado de perfiles laborales coherentes a partir de más de 500,000 ofertas laborales.

4.3.2 Estandarización con ESCOX

En una línea complementaria, enfocada en la estandarización, se encuentra la herramienta open-source ESCOX, presentada por Kavargyris et al. (2025). ESCOX fue diseñada para operacionalizar el mapeo semántico de texto no estructurado contra las taxonomías ESCO e ISCO-08.

Su arquitectura se basa en el uso de un modelo Sentence Transformer pre-entrenado (all-MiniLM-L6-v2) para generar embeddings tanto del texto de entrada como de todas las entidades de ESCO. Posteriormente, calcula la similitud del coseno entre el texto de entrada y cada entidad de la taxonomía, devolviendo aquellas que superan un umbral predefinido (default: 0.6 para skills, 0.55 para occupations).

El sistema ofrece:

- Backend Flask API con endpoints REST

- Matching por cosine similarity contra embeddings precomputados de ESCO/ISCO
- Umbrales ajustables
- Deployment con Docker Compose (Flask + Gunicorn + Nginx)
- GUI no-code para usuarios sin conocimientos técnicos
- API REST para integración en pipelines

En un caso de estudio con 6,500 ofertas laborales de EURES en el dominio de software engineering, ESCOX extrajo aproximadamente 7,400 habilidades y 6,100 ocupaciones. Las skills más frecuentes fueron Java (27.7 %), SQL (19.2 %), DevOps (12.8 %), Work independently (10.1 %), y Python (5.9 %) (Kavargyris et al., 2025).

El valor de ESCOX reside en su practicidad, eficiencia y su naturaleza de código abierto, ofreciendo una solución accesible para la estandarización de habilidades. Sin embargo, sus propios autores reconocen la limitación de su enfoque: al ser un método basado en embeddings pre-entrenados sin fine-tuning, su precisión es inherentemente menor que la de modelos más avanzados y especializados (Kavargyris et al., 2025).

4.3.3 Embeddings multilingües para clasificación ocupacional

El trabajo de Kavas et al. (2024) abordó el reto de clasificar ofertas laborales multilingües (español, italiano) contra la taxonomía ESCO en inglés, enfrentando el problema de solapamiento de ocupaciones y ambigüedad de skills.

Propusieron un modelo híbrido que combina:

- **Embeddings multilingües (E5-large)**: Para recuperar las definiciones de ocupaciones ESCO más cercanas por similitud coseno (top 30).
- **Retrieval-Augmented Generation (RAG)**: Para enriquecer al LLM con contexto específico y reducir alucinaciones.
- **LLMs optimizados (Llama-3 8B)**: Con Chain-of-Thought + DSPy para seleccionar títulos ocupacionales finales dentro de un conjunto restringido.

Los resultados sobre 200 ofertas reales (100 IT, 100 ES) de InfoJobs mostraron:

- Llama-3-8B (CoT optimizado) $\rightarrow$ Precisión@5 = 0.32 (IT), 0.28 (ES); Recall@5 = 0.76 (IT), 0.72 (ES)

- Embeddings E5-large → Recall@10 = 0.88 (IT), 0.92 (ES)
- Superación amplia de baselines (SkillGPT, MNLI) (Kavas et al., 2024)

Este trabajo validó que el enfoque híbrido embeddings → LLM → clasificación es efectivo para contextos multilingües, demostrando que los embeddings multilingües son clave para recall alto, mientras que los LLMs refinan precisión (Kavas et al., 2024).

En trabajos posteriores del mismo grupo, Kavas et al. (2025) extendieron este enfoque hacia la extracción multilingüe de habilidades para job matching en knowledge graphs, integrando extracción con NER, embeddings E5 y LLMs con RAG, demostrando la viabilidad de pipelines completos para matching de vacantes y candidatos en múltiples idiomas (Kavas et al., 2025).

4.4 Análisis Comparativo y Valor Agregado de la Solución Propuesta

El análisis del contexto revela un panorama de investigación rico pero fragmentado, donde ninguna solución existente abordaba de manera integral los desafíos del mercado laboral tecnológico en América Latina. La Tabla 4.1 resume las características principales de las líneas de trabajo analizadas.

Tabla 4.1: Comparación de enfoques en el estado del arte

Enfoque	**Ventajas**	**Limitaciones**	**Referencias**
Enfoques Regionales (Léxicos)	- Validación de web scraping - Datos de alta frecuencia - Contexto local	- Dependencia léxica - Escalabilidad limitada - No captura semántica	(Aguilera & Méndez, 2018; Martínez Sánchez, 2024; Rubio Arrubla, 2025)
LLMs Prompting	- Flexibilidad - Sin necesidad de entrenamiento - Captura contexto complejo	- Inconsistencia de salida - Alucinaciones - F1 bajo (17-27 %)	(Nguyen et al., 2024)
LLMs Fine-tuned	- SOTA en F1 (64.8 %) - Salidas estructuradas - Auditabilidad	- Requiere datasets anotados - Solo en inglés - Costoso computacionalmente	(Herandi et al., 2024; Zhang et al., 2022)
Pipelines Semánticos	- Descubrimiento no supervisado - Identificación de perfiles - Metodología validada	- No incluye LLMs - Limitado a extracción regex inicial	(Lukauskas et al., 2023)
Herramientas de Estandarización	- Open-source - Integración ESCO/ISCO - Fácil de usar	- Precisión limitada - No captura skills emergentes	(Kavargyris et al., 2025)

4.4.1 Síntesis estratégica de metodologías

Ante esta realidad, el sistema desarrollado en este proyecto no se posicionó como una alternativa incremental, sino como una síntesis estratégica que articuló las fortalezas de las distintas líneas de investigación para crear una solución metodológicamente superior y contextualmente relevante.

El proyecto partió de los aprendizajes de los estudios regionales, adoptando su enfoque en la recolección de datos masivos a través de web scraping como una fuente válida y de alta frecuencia para caracterizar el mercado (Aguilera & Méndez, 2018; Martínez Sánchez, 2024; Rubio Arrubla, 2025). Sin embargo, reemplazó conscientemente su análisis léxico, propenso a errores y de limitada profundidad, con la potencia semántica e inferencial de los Large Language Models (LLMs).

Para ello, se inspiró en la investigación internacional de vanguardia, tanto en la exploración del prompting para manejar frases ambiguas (Nguyen et al., 2024), como en la implementación de técnicas de fine-tuning para alcanzar un rendimiento de última generación (Herandi et al., 2024).

Además, la solución no se detuvo en la simple extracción de habilidades, sino que buscó estructurar el conocimiento descubierto. Para lograrlo, integró el robusto pipeline de análisis no supervisado validado empíricamente por Lukauskas et al. (2023), combinando embeddings, UMAP y HDBSCAN para la identificación automática de clústeres de competencias.

4.4.2 Adaptación al contexto latinoamericano

Crucialmente, todo el sistema fue diseñado desde su concepción para adaptarse a la realidad lingüística y de datos de América Latina, un vacío metodológico dejado por la investigación internacional, que se ha centrado casi exclusivamente en datasets en inglés (Herandi et al., 2024).

Esta adaptación incluyó:

- Uso de embeddings multilingües (E5) para manejar el “Spanglish” técnico
- Validación con datos de Colombia, México y Argentina
- Integración con taxonomías internacionales (ESCO) pero adaptadas al contexto regional
- Manejo de la informalidad y variabilidad en la redacción de ofertas laborales

4.4.3 Arquitectura comparativa dual: valor agregado

Finalmente, el valor agregado más significativo del proyecto residió en su arquitectura comparativa (Pipeline A vs. Pipeline B). Este diseño dual no solo permitió aprovechar lo

mejor de los métodos tradicionales y de los LLMs, sino que introdujo un marco de validación empírica que aporta un rigor científico del que carecen muchas aplicaciones prácticas.

Al contrastar sistemáticamente un método transparente y auditable (Pipeline A: NER + Regex + ESCO) contra un modelo semántico avanzado (Pipeline B: LLMs), el sistema no solo generó resultados, sino que también proveyó una medida de la fiabilidad y el valor agregado de cada enfoque, constituyendo una contribución novedosa y completa al campo de los observatorios laborales automatizados.

Capítulo 5

DISEÑO DE LA SOLUCIÓN

El diseño del observatorio de demanda laboral se fundamentó en una arquitectura modular de pipeline lineal, donde cada componente procesa datos de forma secuencial y almacena sus resultados en una base de datos centralizada. Esta arquitectura garantiza la trazabilidad, la escalabilidad y la capacidad de replicar o reejecutar cualquier etapa del proceso de manera independiente.

5.1 Contexto del Sistema

La Figura 5.1 presenta el diagrama de contexto del observatorio, mostrando los actores externos, sistemas externos y componentes internos principales. Este diagrama, basado en el modelo C4, proporciona una vista de alto nivel de las fronteras del sistema y sus interacciones con el entorno.

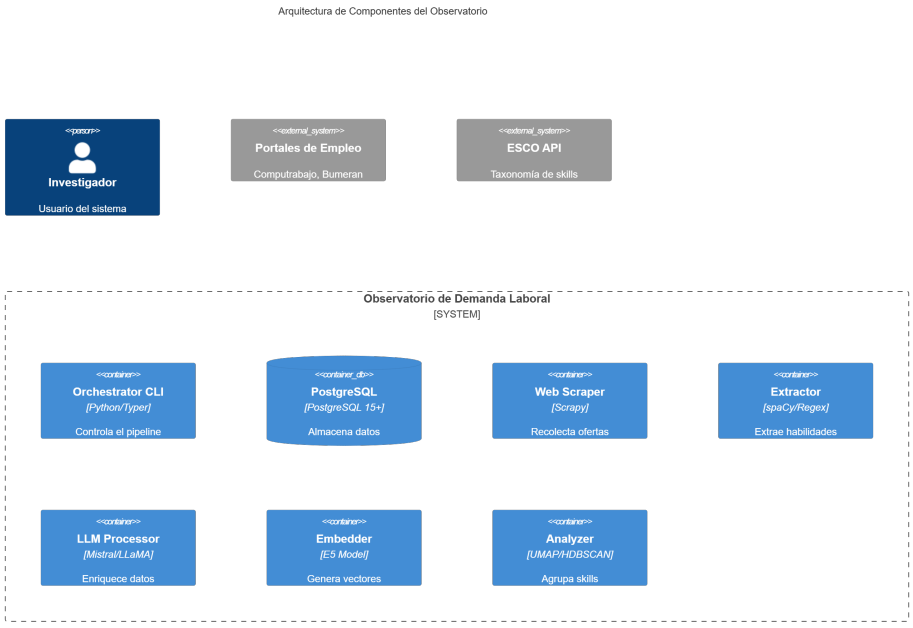

Figura 5.1: Diagrama de Contexto del Observatorio de Demanda Laboral

El sistema interactúa con tres entidades externas principales:

- **Investigador**: Usuario principal del sistema que inicia el proceso de scraping, configura parámetros de análisis y consume los reportes generados.
- **Portales de Empleo**: Fuentes de datos externas (Computrabajo, Bumeran, ElEmpleo) que publican ofertas laborales en formato HTML.
- **ESCO API**: Taxonomía europea de habilidades y competencias utilizada para normalizar y clasificar las habilidades extraídas.

Internamente, el sistema está compuesto por siete componentes principales que operan de forma coordinada: Orchestrator CLI (control del pipeline), PostgreSQL (almacenamiento), Web Scraper (adquisición), Extractor (NLP), LLM Processor (enriquecimiento), Embedder (vectorización) y Analyzer (clustering y visualización).

5.2 Arquitectura General del Sistema

5.2.1 Pipeline Lineal de 8 Etapas

El sistema se diseñó como un pipeline secuencial compuesto por ocho módulos especializados que transforman progresivamente los datos brutos en conocimiento estructurado sobre el mercado laboral:

1. **Módulo de Scraping (Scrapy)**: Recolección automatizada de ofertas laborales desde portales web.
2. **Módulo de Extracción (NER + Regex)**: Identificación de habilidades explícitas mediante reconocimiento de entidades y patrones.
3. **Módulo LLM (Mistral/LLaMA)**: Enriquecimiento semántico y detección de habilidades implícitas.
4. **Módulo de Embeddings (E5 Multilingüe)**: Generación de representaciones vectoriales de habilidades.
5. **Módulo de Reducción Dimensional (UMAP)**: Proyección a espacios de baja dimensionalidad.
6. **Módulo de Clustering (HDBSCAN)**: Agrupamiento no supervisado de habilidades.
7. **Módulo de Visualización**: Generación de gráficos y páginas web estáticas.
8. **Módulo de Reportes**: Exportación de resultados en formato PDF, PNG y CSV.

5.2.2 Representación Visual de la Arquitectura

La arquitectura del observatorio se presenta desde tres perspectivas complementarias que permiten comprender el sistema en su totalidad: la vista modular detallada, la vista por capas de responsabilidad y la vista de transformación de datos.

Vista Modular: Pipeline de 8 Etapas

La Figura 5.2 presenta la arquitectura modular completa del sistema, detallando cada una de las ocho etapas del pipeline con sus tecnologías específicas, funciones, entradas, salidas y

mecanismos de almacenamiento. Esta vista es fundamental para comprender la responsabilidad individual de cada módulo y cómo estos se integran para formar el sistema completo.

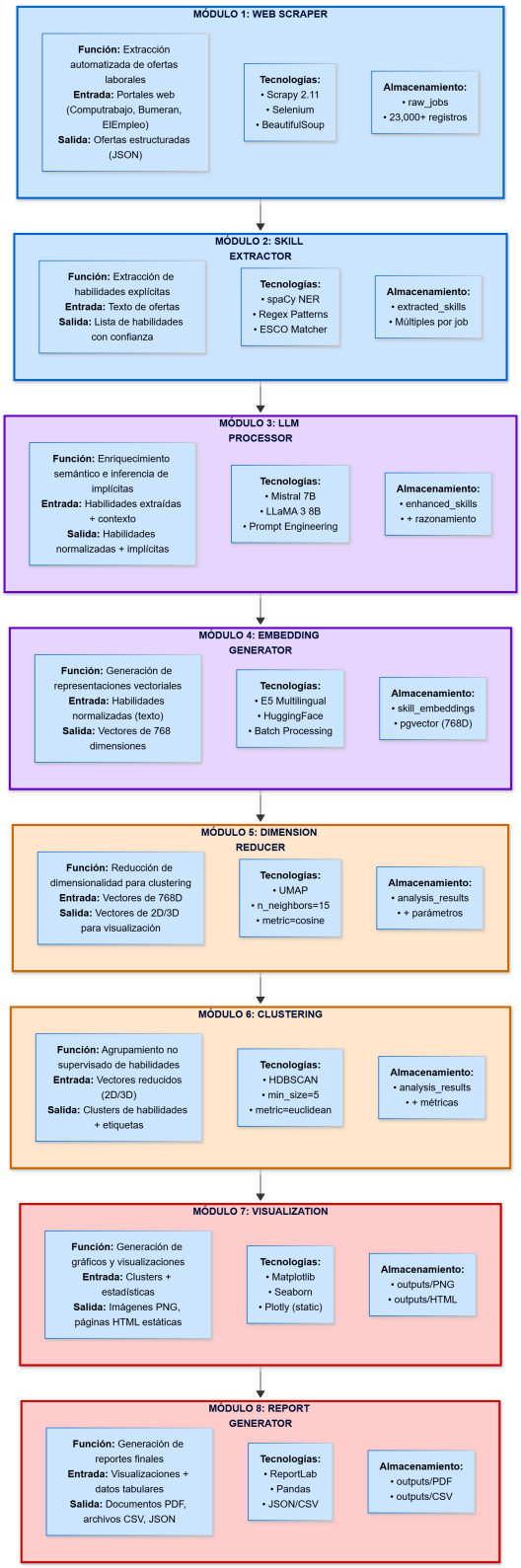

Figura 5.2: Arquitectura Modular Completa del Observatorio - Pipeline de 8 Etapas

Cada módulo del pipeline opera de forma autónoma y puede ser ejecutado independientemente, lo que facilita el desarrollo incremental, las pruebas unitarias y la depuración. Los módulos 1-2 se enfocan en la adquisición y extracción inicial de datos; los módulos 3-4 realizan el enriquecimiento semántico mediante inteligencia artificial; los módulos 5-6 ejecutan el análisis no supervisado; y finalmente, los módulos 7-8 generan las salidas consumibles por usuarios finales.

Vista por Capas: Separación de Responsabilidades

La Figura 5.3 reorganiza los componentes del sistema en siete capas lógicas, mostrando cómo se distribuyen las responsabilidades y cómo fluye el control y los datos entre capas. Esta representación es especialmente útil para comprender la arquitectura desde una perspectiva de ingeniería de software y para identificar las dependencias entre componentes.

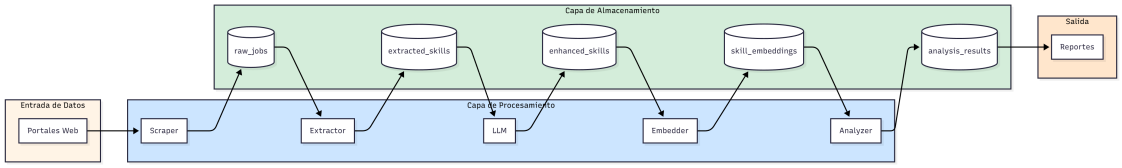

Figura 5.3: Arquitectura en Capas del Sistema - Vista de Separación de Responsabilidades

Las capas 1-5 representan el flujo vertical de procesamiento de datos (de arriba hacia abajo), mientras que la capa 6 (Persistencia) actúa como el repositorio central al que todas las capas superiores acceden para lectura y escritura. La capa 7 (Orquestación) coordina transversalmente la ejecución de todas las demás capas, implementando el patrón de orquestador que controla el ciclo de vida completo del sistema.

Características clave del flujo de procesamiento:

- **Deduplicación en Scraper (Capa 1)**: El módulo de scraping implementa detección de duplicados mediante hashing SHA-256 del contenido (`content_hash`), evitando el almacenamiento de ofertas repetidas antes de su persistencia.
- **Limpieza de texto en Extractor (Capa 2)**: El Skill Extractor realiza preprocesamiento y normalización del texto (eliminación de caracteres especiales, normalización de espacios, conversión a minúsculas) antes de aplicar técnicas de NER y Regex.
- **Procesamiento paralelo (Capas 2-3)**: Los módulos 2 (Skill Extractor) y 3 (LLM Processor) están diseñados para ejecutarse en paralelo sobre diferentes lotes de ofertas

laborales, maximizando el throughput del sistema mediante procesamiento concurrente.

Vista de Estados: Ciclo de Vida de Jobs

La Figura 5.4 ilustra el ciclo de vida completo que atraviesan las ofertas laborales desde su ingreso al sistema hasta la generación de reportes. Este diagrama de estados muestra las transiciones entre las diferentes etapas de procesamiento, incluyendo los caminos de manejo de errores y reintentos.

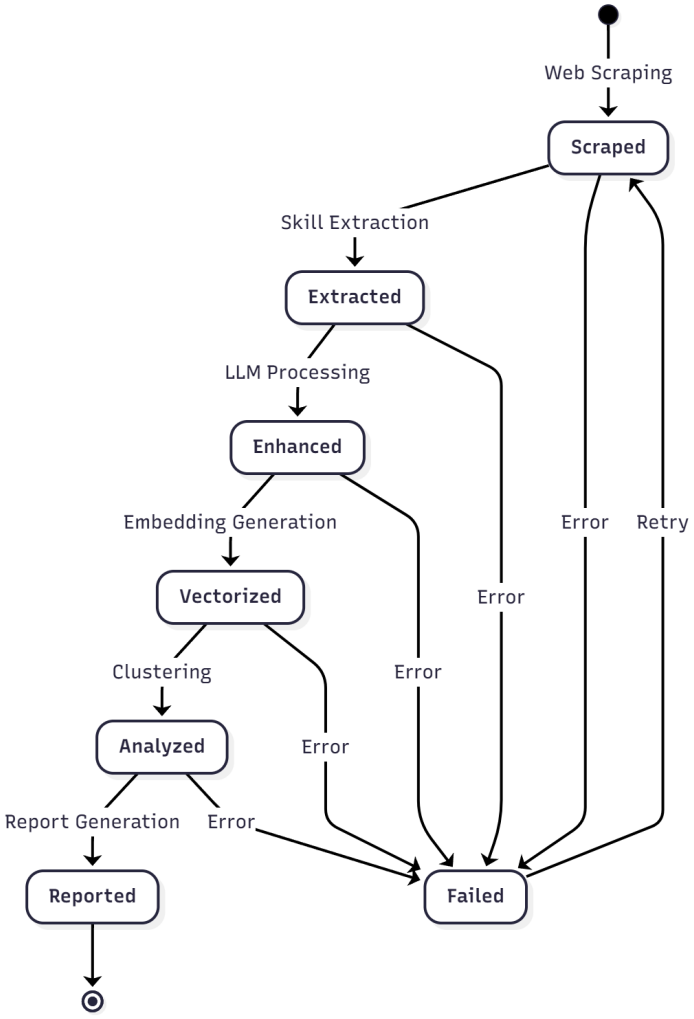

Figura 5.4: Diagrama de Estados del Procesamiento de Ofertas Laborales

Como se observa, los datos atraviesan seis estados principales: **Scraped** → **Extracted** → **Enhanced** → **Vectorized** → **Analyzed** → **Reported**. En cada transición, el sistema puede

detectar errores que llevan al estado **Failed**, desde donde se pueden aplicar reintentos automáticos. Esta arquitectura de estados garantiza la trazabilidad completa y la capacidad de recuperación ante fallos en cualquier etapa del pipeline.

5.3 Diseño de la Base de Datos

La arquitectura de datos se sustenta en PostgreSQL 15+, seleccionado por su robustez, soporte JSON nativo y capacidad de extensión mediante pgvector para operaciones de similitud vectorial.

5.3.1 Esquema de la Base de Datos

La base de datos está compuesta por seis tablas principales que capturan el flujo completo del pipeline:

Tabla: raw_jobs

Almacena las ofertas laborales tal como fueron extraídas de los portales web:

```
CREATE TABLE raw_jobs (
    job_id UUID PRIMARY KEY,
    portal VARCHAR(50) NOT NULL,
    country CHAR(2) NOT NULL,
    url TEXT NOT NULL,
    title TEXT NOT NULL,
    company TEXT,
    location TEXT,
    description TEXT NOT NULL,
    requirements TEXT,
    salary_raw TEXT,
    contract_type VARCHAR(50),
    remote_type VARCHAR(50),
    posted_date DATE,
    scraped_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    content_hash VARCHAR(64) UNIQUE,
    is_processed BOOLEAN DEFAULT FALSE
```

```
);
```

Campos clave:

- `job_id`: Identificador único generado con UUID v4
- `portal`: Origen de la oferta (computrabajo, bumeran, elempleo)
- `country`: Código ISO 3166-1 alpha-2 (CO, MX, AR)
- `content_hash`: Hash SHA-256 del contenido para detección de duplicados
- `is_processed`: Bandera de control para procesamiento incremental

Tabla: extracted_skills

Contiene las habilidades identificadas mediante técnicas de NER y expresiones regulares:

```
CREATE TABLE extracted_skills (
    extraction_id UUID PRIMARY KEY,
    job_id UUID REFERENCES raw_jobs(job_id),
    skill_text TEXT NOT NULL,
    skill_type VARCHAR(50),
    extraction_method VARCHAR(50),
    confidence_score FLOAT,
    source_section VARCHAR(50),
    span_start INTEGER,
    span_end INTEGER,
    esco_uri TEXT,
    extracted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Campos clave:

- `extraction_method`: Indica el método de extracción (ner, regex, esco_match)
- `confidence_score`: Nivel de confianza de la extracción (0-1)
- `source_section`: Sección de origen (title, description, requirements)
- `span_start/span_end`: Posición del span en el texto original
- `esco_uri`: Enlace a la taxonomía ESCO cuando existe correspondencia

Tabla: enhanced_skills

Almacena las habilidades enriquecidas por el procesamiento LLM:

```
CREATE TABLE enhanced_skills (
    enhancement_id UUID PRIMARY KEY,
    job_id UUID REFERENCES raw_jobs(job_id),
    original_skill_text TEXT,
    normalized_skill TEXT NOT NULL,
    skill_type VARCHAR(50),
    esco_concept_uri TEXT,
    esco_preferred_label TEXT,
    llm_confidence FLOAT,
    llm_reasoning TEXT,
    is_duplicate BOOLEAN DEFAULT FALSE,
    duplicate_of_id UUID,
    enhanced_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    llm_model VARCHAR(100)
);
```

Campos clave:

- `normalized_skill`: Versión normalizada de la habilidad según ESCO
- `skill_type`: Tipo de habilidad (explicit, implicit, normalized)
- `llm_confidence`: Confianza del modelo LLM en la inferencia
- `llm_reasoning`: Justificación del modelo para habilidades implícitas
- `is_duplicate`: Bandera para habilidades duplicadas identificadas

Tabla: skill_embeddings

Contiene las representaciones vectoriales de las habilidades:

```
CREATE TABLE skill_embeddings (
    embedding_id UUID PRIMARY KEY,
    skill_text TEXT UNIQUE NOT NULL,
```

```
    embedding vector(768) NOT NULL,
    model_name VARCHAR(100) NOT NULL,
    model_version VARCHAR(50),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE INDEX ON skill_embeddings
USING ivfflat (embedding vector_cosine_ops)
WITH (lists = 100);
```

Características:

- Utiliza la extensión `pgvector` para almacenar vectores de 768 dimensiones
- Implementa índice IVFFlat para búsquedas de similitud eficientes
- Los vectores son generados con el modelo E5 multilingüe
- Soporte para operaciones de distancia coseno

Tabla: analysis_results

Almacena los resultados de análisis de clustering y tendencias:

```
CREATE TABLE analysis_results (
    analysis_id UUID PRIMARY KEY,
    analysis_type VARCHAR(50),
    country CHAR(2),
    date_range_start DATE,
    date_range_end DATE,
    parameters JSONB,
    results JSONB,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Campos clave:

- `analysis_type`: Tipo de análisis (clustering, trends, profile)

- `parameters`: Configuración del análisis en formato JSON
- `results`: Resultados estructurados en formato JSON
- Uso de tipo de datos JSONB para flexibilidad y eficiencia en consultas

Tabla: esco_skills

Contiene la taxonomía ESCO completa con más de 13,000 habilidades:

```
CREATE TABLE esco_skills (
    esco_uri TEXT PRIMARY KEY,
    preferred_label_es TEXT NOT NULL,
    preferred_label_en TEXT,
    alt_labels TEXT[],
    skill_type VARCHAR(50),
    description TEXT,
    skill_reuse_level VARCHAR(50),
    last_updated TIMESTAMP
);
```

Características:

- Integración completa de la taxonomía ESCO v1.1.0
- Soporte multilingüe (español e inglés)
- Almacenamiento de etiquetas alternativas como array
- Metadatos sobre el nivel de reutilización de la habilidad

5.3.2 Modelo Entidad-Relación

La Figura 5.5 presenta el modelo entidad-relación completo de la base de datos, mostrando las cinco tablas principales del pipeline y sus relaciones con la tabla de referencia ESCO. Este diagrama ilustra claramente el flujo de datos entre tablas y las dependencias mediante llaves foráneas.

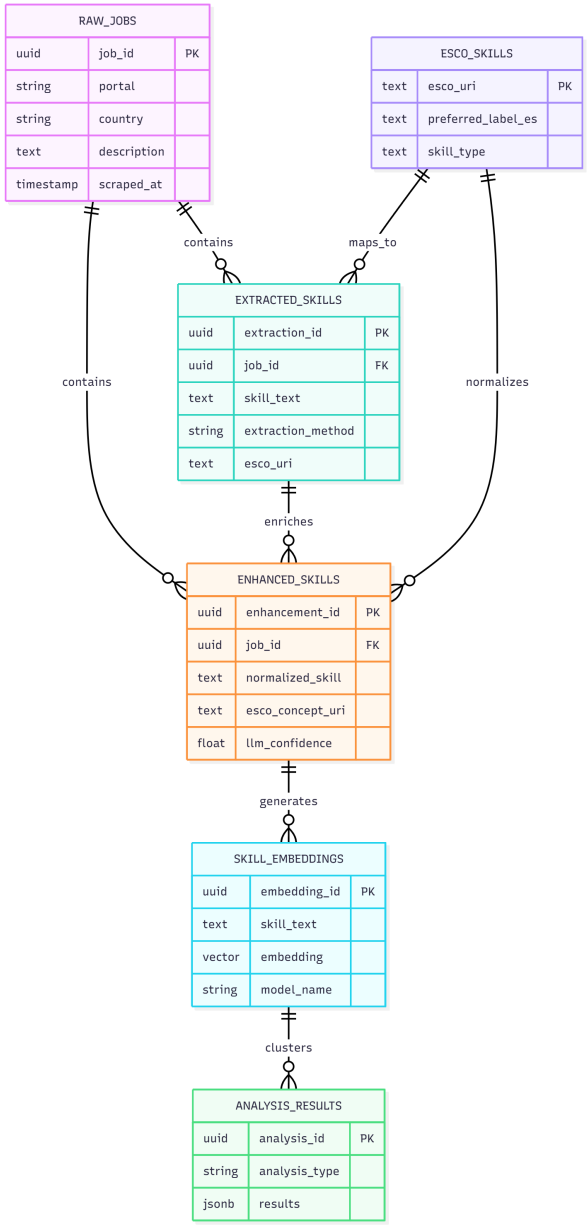

Figura 5.5: Diagrama Entidad-Relación de la Base de Datos del Observatorio

El modelo de datos sigue una arquitectura de pipeline lineal donde cada tabla representa una etapa de procesamiento:

- `raw_jobs` es el punto de entrada y contiene referencias (FK) en todas las tablas derivadas
- `extracted_skills` enriquece (*enriches*) a `enhanced_skills`

- `enhanced_skills` genera (*generates*) vectores en `skill_embeddings`
- `skill_embeddings` alimenta (*clusters*) los resultados en `analysis_results`
- `esco_skills` actúa como tabla de referencia normalizada mediante mapeos desde `extracted_skills` y `enhanced_skills`

Esta arquitectura garantiza integridad referencial y permite trazabilidad completa desde cualquier resultado de análisis hasta la oferta laboral original.

5.4 Especificación de Módulos

5.4.1 Módulo 1: Web Scraper

Propósito: Recolección automatizada de ofertas laborales desde portales de empleo en Colombia, México y Argentina.

Tecnologías:

- **Scrapy 2.11**: Framework asíncrono para scraping a gran escala
- **Selenium + ChromeDriver**: Para portales con contenido dinámico
- **BeautifulSoup 4.12**: Parsing y extracción de HTML

Características clave:

- Scraping concurrente con límites de tasa por portal
- Rotación de user-agents y delays adaptativos
- Detección de duplicados mediante hashing de contenido
- Reintentos con backoff exponencial ante fallos
- Manejo de paginación automática

Portales soportados:

- Computrabajo (CO, MX, AR)
- Bumeran (MX, AR)

- ElEmpleo (CO)
- InfoJobs (MX)
- OCC Mundial (MX)
- ZonaJobs (AR)

5.4.2 Módulo 2: Skill Extractor

Propósito: Extracción de habilidades explícitas mediante técnicas de NLP.
Componentes:

1. **NER Extractor**: Utiliza spaCy con el modelo `es_core_news_lg` y un EntityRuler personalizado poblado con la taxonomía ESCO completa para reconocer entidades de tipo skill/technology.
2. **Regex Patterns**: Conjunto de expresiones regulares especializadas para capturar tecnologías con nomenclatura específica (ej. “Node.js”, “React.js”, “Python 3.x”).
3. **ESCO Matcher**: Módulo de mapeo que normaliza las habilidades extraídas contra la taxonomía ESCO mediante:
   - Coincidencia exacta (case-insensitive)
   - Coincidencia difusa con umbral de similitud
   - Búsqueda en etiquetas alternativas

Pipeline de procesamiento:

```
1. Concatenar: title + description + requirements
2. Preprocesamiento: limpieza y normalización de texto
3. Extracción NER: identificar entidades con EntityRuler
4. Extracción Regex: aplicar patrones de tecnologías
5. Deduplicación: eliminar menciones repetidas
6. Mapeo ESCO: normalizar contra taxonomía
7. Persistencia: almacenar en extracted_skills
```

5.4.3 Módulo 3: LLM Processor

Propósito: Enriquecimiento semántico de habilidades y detección de competencias implícitas.

Modelos soportados:

- **Mistral 7B Instruct**: Modelo local mediante llama-cpp-python (cuantización Q4)
- **LLaMA 3 8B**: Alternativa con mejor rendimiento en español
- **OpenAI GPT-4**: Fallback opcional mediante API

Tareas del módulo:

1. **Deduplicación inteligente**: Identificar variantes de la misma habilidad ("React", "React.js", "ReactJS")
2. **Inferencia de habilidades implícitas**: Detectar competencias no mencionadas explícitamente pero requeridas por el contexto del cargo
3. **Normalización con ESCO**: Mapear todas las habilidades a conceptos ESCO con confianza
4. **Razonamiento explicable**: Generar justificaciones para habilidades implícitas

Prompt Engineering: Se diseñaron prompts específicos para el contexto latinoamericano que incluyen:

- Instrucciones para manejar "Spanglish" (términos técnicos en inglés en contexto español)
- Few-shot examples con ofertas laborales reales de la región
- Restricción a la taxonomía ESCO para salidas estructuradas
- Solicitud de formato JSON con campos definidos

Configuración experimental: El sistema permite ejecutar el pipeline LLM de dos formas:

1. **Comparación multi-modelo**: Procesar las mismas ofertas con Mistral 7B, LLaMA 3 8B y GPT-4 para evaluar diferencias en rendimiento, consistencia y calidad de extracción.

2. **Producción con LLaMA**: Usar exclusivamente LLaMA 3 8B como modelo estándar por su balance entre rendimiento y velocidad de inferencia en español.

5.4.4 Módulo 4: Embedding Generator

Propósito: Generar representaciones vectoriales semánticas de habilidades.
Modelo: `intfloat/multilingual-e5-base`

- Modelo de embeddings multilingüe de 768 dimensiones
- Entrenado específicamente para similitud semántica
- Soporte nativo para español e inglés en el mismo espacio vectorial

Proceso:

```
1. Cargar modelo E5 desde Hugging Face
2. Preprocesar: añadir prefijo "query: " según especificación E5
3. Generar embeddings por lotes (batch_size=32)
4. Normalizar vectores (L2 normalization)
5. Almacenar en PostgreSQL con pgvector
6. Crear índice IVFFlat para búsquedas eficientes
```

Pipeline A vs Pipeline B - Comparación experimental:

El sistema implementa dos pipelines de procesamiento paralelos para evaluar la efectividad de diferentes enfoques:

- **Pipeline A (NER + Regex + ESCO)**: Extracción basada en reglas (Módulo 2) seguida de generación de embeddings (Módulo 4)
- **Pipeline B (LLM + ESCO)**: Extracción semántica con LLMs (Módulo 3) seguida de generación de embeddings (Módulo 4)

Esta arquitectura dual permite comparar cuantitativamente:

1. Cantidad y calidad de habilidades extraídas por cada método
2. Cobertura de la taxonomía ESCO alcanzada
3. Consistencia de los embeddings generados

4. Rendimiento de clustering y descubrimiento de patrones

Los resultados de ambos pipelines convergen en la capa de embeddings, donde se puede evaluar objetivamente cuál enfoque genera representaciones vectoriales más útiles para análisis downstream.

5.4.5 Módulo 5: Analyzer

Propósito: Descubrir patrones y clústeres de habilidades mediante análisis no supervisado.

Componentes:

Reducción de Dimensionalidad (UMAP)

UMAP (Uniform Manifold Approximation and Projection):

- Reduce los vectores de 768 dimensiones a 2-3 dimensiones
- Preserva tanto estructura local como global
- Parámetros clave:
  - `n_neighbors=15`: Balance entre estructura local y global
  - `min_dist=0.1`: Compactación de puntos cercanos
  - `metric='cosine'`: Métrica de similitud

Clustering (HDBSCAN)

HDBSCAN (Hierarchical Density-Based Spatial Clustering):

- Algoritmo de clustering basado en densidad
- No requiere especificar número de clústeres
- Identifica ruido automáticamente
- Parámetros clave:
  - `min_cluster_size=5`: Tamaño mínimo de clúster válido
  - `min_samples=3`: Puntos mínimos para densidad
  - `metric='euclidean'`: Post-reducción dimensional

Visualización

Generación de gráficos estáticos:

- Scatter plots 2D/3D de clústeres con matplotlib
- Distribuciones de frecuencia de habilidades
- Heatmaps de co-ocurrencia de habilidades
- Gráficos de barras por país y portal

Generación de Reportes

Exportación multi-formato:

- **PDF**: Reportes completos con ReportLab incluyendo:
  - Resumen ejecutivo de hallazgos
  - Estadísticas descriptivas
  - Visualizaciones embebidas
  - Tablas de top skills por clúster
- **PNG**: Imágenes de alta resolución de visualizaciones
- **CSV**: Datos tabulares para análisis externo
- **JSON**: Resultados estructurados para integración

5.5 Decisiones Técnicas y Justificación

La siguiente tabla resume las decisiones arquitectónicas clave y su fundamentación:

Componente	Tecnología	Justificación
Base de datos	PostgreSQL 15+	Soporte JSON nativo (JSONB), extensión pgvector para vectores, robustez empresarial, licencia libre (PostgreSQL License).

Componente	**Tecnología**	**Justificación**
Taxonomía	ESCO v1.1.0	Cobertura superior en español (13,000+ skills), etiquetas multilingües, amplia representación de habilidades tecnológicas, respaldo institucional de la Comisión Europea.
Framework de scraping	Scrapy 2.11	Arquitectura asíncrona para alto rendimiento, manejo robusto de reintentos y errores, middlewares extensibles, amplia comunidad y documentación.
Modelo NLP	spaCy 3.7 + es_core_news_lg	Mejor modelo disponible para español, soporte de EntityRuler para reglas personalizadas, rendimiento optimizado en CPU.
LLM local	Mistral 7B / LLaMA 3 8B	Ejecución local sin dependencia de APIs externas, buen rendimiento en español post-instrucción, cuantización Q4 para reducir requisitos de memoria (4-5 GB RAM).
Modelo de embeddings	intfloat/multilingual-e5-base	Estado del arte en embeddings multilingües, 768 dimensiones balancean expresividad y eficiencia, soporte nativo para español e inglés en espacio compartido.
Algoritmo de clustering	HDBSCAN	No requiere especificar k, identifica ruido, maneja clústeres de densidades variables, jerárquico permite análisis multinivel.
Reducción dimensional	UMAP	Preserva estructura local y global, superior a t-SNE en escalabilidad y reproducibilidad, parámetros interpretables.
Orquestación	Typer CLI	Interface de línea de comandos tipo Git, validación automática de parámetros, ayuda auto-generada, fácil integración con schedulers.

5.6 Orquestación y Automatización

5.6.1 Flujo de Interacciones entre Componentes

La Figura 5.6 presenta el diagrama de secuencia que ilustra el flujo completo de interacciones entre los componentes del sistema durante un ciclo de ejecución completo, desde el comando inicial del usuario hasta la generación de reportes finales.

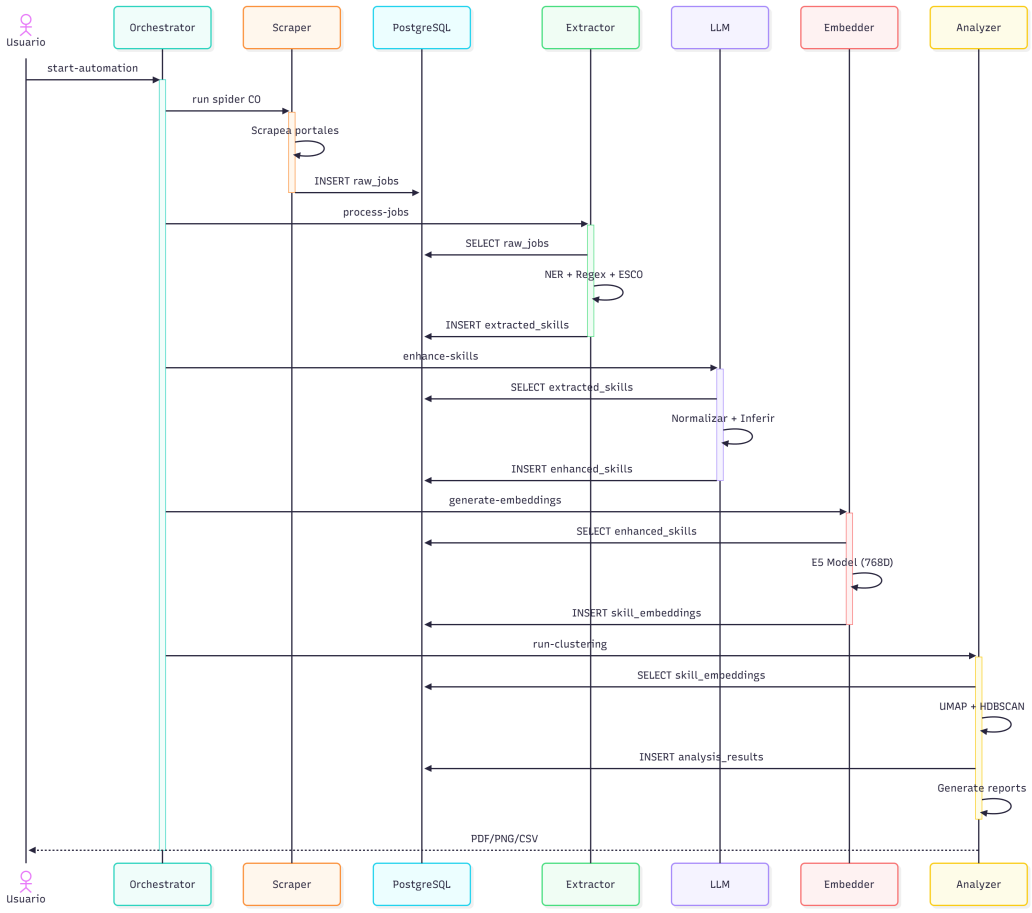

Figura 5.6: Diagrama de Secuencia de Interacciones del Pipeline Completo

El flujo de ejecución sigue estos pasos:

1. El usuario ejecuta `start-automation` en el Orchestrator CLI
2. El Orchestrator lanza el comando `run spider CO` al módulo Scraper
3. El Scraper extrae ofertas de los portales web y las inserta en PostgreSQL (`raw_jobs`)

4. El Orchestrator ejecuta `process-jobs`, que activa el Extractor
5. El Extractor lee ofertas de `raw_jobs`, aplica NER + Regex + ESCO, e inserta en `extracted_skills`
6. El Orchestrator ejecuta `enhance-skills`, que activa el LLM Processor
7. El LLM lee de `extracted_skills`, normaliza e infiere habilidades implícitas, e inserta en `enhanced_skills`
8. El Orchestrator ejecuta `generate-embeddings`, que activa el Embedder
9. El Embedder lee de `enhanced_skills`, genera vectores 768D con E5 Model, e inserta en `skill_embeddings`
10. El Orchestrator ejecuta `run-clustering`, que activa el Analyzer
11. El Analyzer lee de `skill_embeddings`, aplica UMAP + HDBSCAN, genera reportes, e inserta resultados en `analysis_results`
12. El Analyzer retorna archivos PDF/PNG/CSV al usuario

Cada paso incluye confirmaciones bidireccionales entre componentes y la base de datos, garantizando consistencia en cada etapa. El manejo de errores permite reintentos en cualquier punto del flujo sin pérdida de datos.

5.6.2 Orchestrator CLI

El sistema se controla mediante un CLI único implementado con Typer:

```
python -m src.orchestrator <comando> [opciones]
```

Comandos principales:

- `run-once <spider><country>`: Ejecutar scraping único
- `run <spiders><country>`: Ejecutar múltiples spiders
- `process-jobs`: Procesar ofertas pendientes
- `status`: Estado del sistema y estadísticas

- `health`: Métricas de salud del sistema
- `start-automation`: Iniciar sistema automatizado
- `automation-status`: Estado del scheduler
- `list-jobs`: Listar trabajos programados

5.6.3 Sistema de Automatización

Tres componentes coordinan la operación 24/7:

1. **Master Controller**: Coordinador central que gestiona el ciclo de vida del sistema
2. **Intelligent Scheduler**: Basado en APScheduler, programa ejecuciones periódicas de spiders con estrategias adaptativas
3. **Pipeline Automator**: Detecta nuevos trabajos y los procesa automáticamente a través del pipeline de extracción

Programación de spiders:

- Frecuencia configurable por portal (cada 6-12 horas)
- Ventanas de ejecución para minimizar detección
- Priorización dinámica basada en tasa de actualización del portal
- Reintentos automáticos con backoff exponencial

5.7 Métricas de Evaluación

5.7.1 Métricas por Módulo

Scraper

- **Tasa de éxito**: (peticiones exitosas / peticiones totales) × 100
- **Tasa de parseo**: (trabajos parseados / páginas scrapeadas) × 100
- **Tasa de duplicados**: (trabajos duplicados / trabajos totales) × 100
- **Cobertura**: trabajos únicos por portal por país

Extractor

- **Precisión**: habilidades validadas / habilidades extraídas
- **Recall**: habilidades extraídas / habilidades anotadas (gold standard)
- **F1-Score**: media armónica de precisión y recall
- **Tasa de mapeo ESCO**: habilidades mapeadas a ESCO / total habilidades

LLM Processor

- **Tasa de deduplicación**: habilidades deduplicadas / habilidades de entrada
- **Descubrimiento de habilidades implícitas**: habilidades implícitas / total habilidades enriquecidas
- **Éxito de normalización**: habilidades normalizadas con ESCO / total habilidades
- **Tiempo de procesamiento**: tiempo promedio por oferta laboral

Clustering

- **Silhouette Score**: Medida de cohesión y separación de clústeres (-1 a 1, óptimo > 0.5)
- **Davies-Bouldin Index**: Validez de clústeres (menor es mejor)
- **Estabilidad de clústeres**: Consistencia entre ejecuciones con semillas diferentes
- **Cobertura**: trabajos en clústeres / trabajos totales

5.7.2 Métricas del Sistema

Métricas de rendimiento:

- Trabajos procesados por hora
- Tiempo promedio de extracción por trabajo
- Tiempo promedio de procesamiento LLM por trabajo
- Tiempo promedio de generación de embedding por habilidad

- Latencia de búsqueda de similitud vectorial

Métricas de calidad:

- F1 Score de extracción de habilidades
- Cobertura de enriquecimiento LLM
- Silhouette Score de clustering
- Tasa de mapeo a ESCO
- Consistencia inter-anotadores (para subset validado)

Objetivo de escala: El sistema fue diseñado para procesar 600,000 ofertas laborales para la fecha de defensa, con capacidad de crecimiento horizontal mediante:

- Particionamiento de tablas por país y fecha
- Procesamiento por lotes configurable
- Índices optimizados para consultas frecuentes
- Caché de embeddings para habilidades recurrentes

Capítulo 6

SOLUTION DEVELOPMENT

This chapter must describe the process utilized to create the solution and relate it to the methodology that was specified in the proposal. Additionally, this chapter must also show the final product. For instance, showing screenshots and describing their functions.

Capítulo 7

RESULTS

Must present the results of the quality control process, according to what was defined in the methodology. For instance, in a software development project, this section should include the results from standard software testing (unit, functional, system, acceptance, etc.). It is important for them to be consistent with the objective of the project and the methodology used for its development.

This chapter must include an analysis of the results obtained, and conclusions from this analysis.

Capítulo 8

CONCLUSIONS

8.1 Impact Analysis of the Project

Explain the impact of the results of this project in the short, medium, and long term. It should explain the impact in all of the relevant stakeholders.

8.1.1 Impact analysis in systems engineering

8.1.2 Impact analysis in global, economic, environmental, and societal contexts

8.2 Conclusions and Future Work

Explain whether the goals were accomplished and why. Future work that should be explained based on the project results.

IX- REFERENCES

Aguilera, M., & Méndez, S. (2018). Análisis del mercado laboral TI en Argentina mediante web scraping. *Revista Argentina de Informática*.

Azuara, O., et al. (2022). *COVID-19 y el mercado laboral en América Latina: diagnóstico y políticas*. Banco Interamericano de Desarrollo.

Cárdenas Rubio, J., et al. (2015). Análisis del mercado laboral colombiano mediante técnicas de minería de texto. *Revista Colombiana de Computación*.

Echeverría, L., & Rucci, G. (2022). El futuro del trabajo en América Latina y el Caribe: ¿Qué habilidades y educación se necesitan? *Banco Interamericano de Desarrollo*.

Herandi, A., Li, Y., Liu, Z., Hu, X., & Cai, X. (2024, octubre). Skill-LLM: Repurposing general-purpose LLMs for skill extraction. https://doi.org/10.48550/arXiv.2410.12052

Kavargyris, D. C., Georgiou, K., Papaioannou, E., Petrakis, K., Mittas, N., & Angelis, L. (2025). ESCOX: A tool for skill and occupation extraction using LLMs from unstructured text. *Software Impacts*. https://doi.org/10.1016/j.simpa.2025.100772

Kavas, H., Serra-Vidal, M., & Wanner, L. (2024). Enhancing job posting classification with multilingual embeddings and large language models. *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, 440-450. https://doi.org/10.18653/v1/2024.clicit-1.53

Kavas, H., Serra-Vidal, M., & Wanner, L. (2025). Multilingual skill extraction for job vacancy–job seeker matching in knowledge graphs. *Proceedings of the Workshop on Generative AI and Knowledge Graphs (GenAIK)*, 146-155. https://aclanthology.org/2025.genaik-1.15/

Lukauskas, M., Šarkauskaitė, V., Pilinkienė, V., Stundžienė, A., Grybauskas, A., & Bruneckienė, J. (2023). Enhancing skills demand understanding through job ad segmentation using NLP and clustering techniques. *Applied Sciences*, *13*(10), 6119. https://doi.org/10.3390/app13106119

Martínez Sánchez, C. (2024). *Demanda de habilidades digitales en México: un análisis empírico* [Tesis de maestría, UNAM].

Nguyen, T., et al. (2024). Exploring Large Language Models for Skill Extraction from Job Postings. *Proceedings of EMNLP*.

Orozco Puello, C., & Gómez Estrada, H. (2019). Web Scraping: técnicas y aplicaciones para análisis de datos. *Revista Colombiana de Tecnologías de Avanzada*.

Rubio Arrubla, J. F. (2025, junio). *Demanda de habilidades tecnológicas: evidencia desde el mercado laboral colombiano* (Documento CEDE N.º 2025-18). Universidad de los Andes, Centro de Estudios sobre Desarrollo Económico (CEDE).

Zhang, M., Jensen, K. N., Sonniks, S. D., & Plank, B. (2022). SKILLSPAN: Hard and Soft Skill Extraction from English Job Postings. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4962-4984. https://doi.org/10.18653/v1/2022.naacl-main.366

X- APPENDICES

Place in this section of the document a list of all appendices related to the project. Appendices must be downloadable from the website and should be the same as specified in the proposal.