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.

CONTENIDO

RESUMEN

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.

INTRODUCCIÓN

El mercado laboral en América Latina atraviesa una transformación profunda impulsada por la digitalización de la economía. La pandemia de COVID-19 aceleró este proceso, intensificando la demanda de habilidades tecnológicas especializadas y exponiendo las brechas de capital humano en la región [1]. En este escenario, identificar con precisión qué competencias están siendo requeridas por el mercado se ha vuelto estratégico para gobiernos que diseñan políticas de formación, instituciones educativas que ajustan sus programas, y profesionales que planifican su desarrollo de carrera.

Sin embargo, medir esta demanda de manera sistemática presenta desafíos importantes. Los portales de empleo en la región publican vacantes en formatos heterogéneos, sin vocabularios estandarizados, y con alta volatilidad [2]. Las encuestas tradicionales, aunque valiosas, suelen ser retrospectivas y de baja periodicidad, limitando su utilidad para capturar tendencias emergentes [3]. Los estudios previos en países como Colombia, México y Argentina han aportado evidencia empírica importante, pero se han basado principalmente en análisis de frecuencia de términos y clasificaciones manuales, enfoques que no logran capturar la complejidad del lenguaje técnico ni identificar habilidades implícitas en las descripciones de las vacantes [4, 5].

Este proyecto desarrolló un Observatorio de Demanda Laboral para América Latina, un sistema automatizado que recolecta, procesa y analiza ofertas de empleo a escala regional. El observatorio integra múltiples fuentes de información en Colombia, México y Argentina, aplicando técnicas de inteligencia artificial para extraer y estructurar las habilidades demandadas. La solución combina métodos tradicionales de procesamiento de texto con modelos de lenguaje avanzados, permitiendo tanto la identificación de competencias mencionadas explícitamente como la inferencia de aquellas que se derivan del contexto del cargo. El sistema normaliza los resultados contra taxonomías internacionales reconocidas y aplica técnicas de agrupamiento para descubrir perfiles laborales emergentes, generando visualizaciones y reportes que facilitan la comprensión de la dinámica del mercado.

El valor principal de este proyecto reside en tres aspectos. Primero, su escala regional y enfoque multipaís, que permite comparaciones sistemáticas entre mercados. Segundo, su arquitectura dual que combina precisión en la extracción de términos conocidos con capacidad de descubrimiento de competencias nuevas. Tercero, su adaptación específica al contexto latinoamericano, atendiendo las particularidades del español técnico y la mezcla con anglicismos característica del sector tecnológico en la región.

DESCRIPCIÓN GENERAL

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 [2]. 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 [1]. 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 [2].

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 [3-5]. 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 [6, 7]. 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.

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 [3]. 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 [3].

2.1.2 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 [3]. 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 [4]. 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 [5].

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”), ni 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 [2, 8].

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.

2.1.3 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 baseline 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, siguiendo enfoques de vanguardia [9, 10].

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, inspirado en las arquitecturas de herramientas como ESCOX [11]. 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, un enfoque validado por la literatura para el descubrimiento de estructuras en el mercado laboral [8].

2.1.4 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 [3, 4] o en el uso aislado de LLMs [10], 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 (una mejora sobre la propuesta original de ESCOX) 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 [8, 11]. 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 [9], 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 metodologías 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 [9], y la capacidad estructuradora del clustering semántico [8]. 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.

2.2 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.

2.2.1 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.

2.2.2 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.

2.2.3 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.

CONTEXTO DEL PROYECTO

3.1 Antecedentes Conceptuales

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.

3.1.1 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 [12]. 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 [3, 6].

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 (BeautifulSoup, lxml), headless browsers (Playwright, Puppeteer) para contenido dinámico, control de rate limiting con throttling y backoff exponencial, y rotación de user-agents para evitar bloqueos.

La implementación debe seguir principios éticos y legales: respeto del archivo `robots.txt`, delays entre peticiones, registro de fuentes con sellos de tiempo, validación de datos extraídos y monitoreo de cambios en la estructura del DOM.

3.1.2 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.

El preprocesamiento es fundamental para estandarizar los datos textuales. La Tokenización consiste en segmentar el texto en unidades mínimas o “tokens” (generalmente palabras o signos de puntuación) [10], transformando cadenas continuas en secuencias discretas procesables. La Lematización reduce las palabras a su forma base o raíz gramatical, permitiendo agrupar variaciones morfológicas

(por ejemplo, “programar”, “programando” y “programado” se unifican bajo el lema “programar”) [2].

Con el texto limpio y normalizado, el núcleo del desafío consiste en la extracción de habilidades mediante un enfoque híbrido. Las Expresiones Regulares (Regex) permiten identificar secuencias de texto específicas con formatos predecibles [8], siendo efectivas para capturar tecnologías con nomenclaturas estandarizadas. El Reconocimiento de Entidades Nombradas (NER) es una técnica de NLP diseñada para identificar y clasificar entidades como habilidades y competencias [9], permitiendo reconocer habilidades en contextos gramaticales complejos.

3.1.3 Large Language Models (LLMs)

Para superar las limitaciones de la extracción de menciones explícitas, el proyecto incorpora Large Language Models (LLMs). Estos modelos de lenguaje a gran escala, como GPT o Llama 3, poseen capacidades de razonamiento contextual que permiten abordar desafíos más complejos [9].

A través del Prompt Engineering, es posible guiar a los LLMs para realizar tareas de enriquecimiento semántico: distinción entre habilidades explícitas e implícitas [10], normalización de variantes terminológicas, clasificación según taxonomías predefinidas y generación de salidas estructuradas en formatos como JSON.

Existen diferentes modalidades de aplicación: zero-shot learning (sin ejemplos previos), few-shot learning (con algunos ejemplos en el prompt) [10] y fine-tuning (re-entrenamiento sobre datasets específicos) [9, 13].

3.1.4 Embeddings Semánticos y Representación Vectorial

Las habilidades extraídas deben representarse de forma que permita su análisis cuantitativo. Los Embeddings Semánticos son representaciones vectoriales en un espacio de alta dimensionalidad donde la distancia entre vectores refleja la similitud semántica entre textos [14]. Esto permite capturar relaciones semánticas complejas, realizar búsquedas por similitud eficientemente y agrupar habilidades relacionadas.

Dado que las ofertas laborales en América Latina 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 [2, 14]. Modelos populares incluyen E5-large, Sentence Transformers basados en BERT y multilingual-e5-base.

3.1.5 Análisis No Supervisado: UMAP y HDBSCAN

Para descubrir patrones y estructuras latentes, se aplica un pipeline de análisis no supervisado. Debido a que los embeddings son vectores de muy alta dimensionalidad (768 dimensiones), lo que genera la “maldición de la dimensionalidad”, se aplica UMAP (Uniform Manifold Approximation and

Projection), un algoritmo no lineal que reduce dimensiones preservando la estructura local y global, superior a métodos lineales como PCA [8].

Sobre los datos reducidos se aplica HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise), un algoritmo de clustering basado en densidad. A diferencia de K-Means, HDBSCAN no requiere especificar el número de clústeres, identifica grupos de formas arbitrarias, separa puntos que no pertenecen a ningún grupo como "ruido" y funciona con clústeres de densidades variables [8]. Esta secuencia metodológica permite la identificación automática de "ecosistemas de habilidades" y perfiles laborales emergentes.

3.1.6 Taxonomías Estandarizadas: ESCO e ISCO

Para asegurar la comparabilidad y estandarización de resultados, el sistema integra taxonomías internacionales. ESCO (European Skills, Competences, Qualifications and Occupations) es una taxonomía multilingüe desarrollada por la Comisión Europea que clasifica más de 13,000 habilidades y conocimientos de manera jerárquica [11]. ISCO-08 (International Standard Classification of Occupations) es un estándar internacional de la OIT para clasificar ocupaciones, proporcionando un marco común para comparar datos ocupacionales entre países. Adicionalmente, se toma como referencia O*NET, un portal del Departamento de Trabajo de EE. UU. que reporta habilidades de alta demanda [3].

3.2 Análisis del Contexto

El desafío de extraer, analizar y comprender la demanda de habilidades a partir de 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 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, agrupadas en tres grandes líneas de trabajo: (1) enfoques regionales en América Latina basados en análisis léxico y reglas manuales; (2) la frontera de la extracción con Large Language Models (LLMs) mediante prompting y fine-tuning; y (3) pipelines semánticos con descubrimiento no supervisado mediante embeddings y clustering.

El siguiente análisis demostrará que ninguna de estas líneas, de forma aislada, resolvía los desafíos metodológicos, geográficos y lingüísticos del mercado laboral tecnológico en América Latina. Esta fragmentación justificó la necesidad de una solución sintética y adaptada.

3.2.1 Enfoques Regionales: Caracterización del Mercado con Métodos Léxicos

La primera línea de trabajo comprende estudios pioneros en América Latina que validaron el uso de portales de empleo en línea como fuente de datos, pero emplearon metodologías de procesamiento de texto basadas en análisis léxico, frecuencias de términos y reglas manuales.

El estudio más completo fue el de Rubio Arrubla (2024) para el mercado colombiano [3]. 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: las vacantes tecnológicas aumentaron 50 % en 18 meses post-pandemia. Se observó una caída en la demanda de herramientas ofimáticas tradicionales como Excel (35.8 % en 2018 a 17.4 % en 2023) y un surgimiento exponencial de tecnologías especializadas como NoSQL (12.3 %), Django (5.5 %) y React (5.3 %) para 2023 [3].

Metodológicamente, implementó una tipología propia de habilidades y clasificó vacantes mediante emparejamiento de texto basado en n-gramas y similitud contra la Clasificación Internacional Uniforme de Ocupaciones (CIUO). Sin embargo, su dependencia de la coincidencia léxica fue una limitación: el método perdía eficiencia a medida que aumentaban las palabras en los títulos, al no capturar el contexto general [3].

De forma análoga, Aguilera y Méndez (2018) para Argentina extrajeron datos de ZonaJobs y Bumeran mediante análisis de frecuencias y bigramas [4]. Para estandarizar el vocabulario informal, construyeron una lista de palabras clave semi-manual, limitando la escalabilidad y adaptabilidad a nuevas tecnologías.

Para México, Martínez Sánchez (2024) combinó datos de encuestas oficiales con scraping, basándose en frecuencia de términos y tipología manual para segmentar habilidades [5].

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

El Banco Interamericano de Desarrollo (BID) señaló la falta de pipelines modernos y automatizados en la región, destacando que la mayoría todavía se basa en reglas fijas o mapeos manuales sin incorporar embeddings ni NLP avanzado [2]. Esta constatación institucional refuerza la conclusión de un vacío sistémico: la ausencia de una solución que superara los enfoques léxicos para proporcionar análisis semántico, dinámico y escalable.

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

Paralelamente, una segunda línea de investigación a nivel internacional ha explorado el uso de LLMs para superar las limitaciones de los métodos léxicos, representando la frontera del estado del arte en extracción semántica.

Nguyen et al. (2024) investigaron el uso de LLMs de propósito general (GPT-3.5, GPT-4) en modalidad prompting sin re-entrenamiento (few-shot learning) [10]. Experimentaron con dos formatos de salida: extracción directa ("EXTRACTION-STYLE") y etiquetado ("NER-STYLE"). Aunque los LLMs no igualaron la precisión de modelos supervisados tradicionales, demostraron capacidad superior para interpretar frases sintácticamente complejas. Sin embargo, el estudio advirtió sobre limitaciones: inconsistencia en formatos de salida, riesgo de "alucinaciones" (entidades no reales) y rendimiento cuantitativo inferior (F1-score entre 17.8 % y 27.8 %) [10].

Tomando estas limitaciones como punto de partida, Herandi et al. (2024) representaron la siguiente evolución: el fine-tuning específico de un LLM [9]. Ajustaron el modelo LLaMA 3 8B utilizando el dataset SkillSpan [13], diseñando un formato de salida estructurado en JSON que extraía la habilidad y su contexto textual. Este enfoque alcanzó el estado del arte (SOTA) con F1-score total de 64.8 % (skills: 54.3 %, knowledge: 74.2 %), superior a modelos supervisados previos y LLMs mediante prompting [9]. El método garantizó consistencia y auditabilidad, resolviendo problemas prácticos de los LLMs.

A pesar de su sofisticación técnica, estos estudios comparten una limitación crucial: fueron desarrollados y validados casi exclusivamente sobre datasets en idioma inglés. El trabajo de Herandi et al. (2024) se fundamentó en SkillSpan, que contiene únicamente ofertas en inglés [9]. Esta dependencia evidenció un vacío geográfico y lingüístico en la aplicación de técnicas de NLP avanzadas para el análisis del mercado laboral.

Si bien los LLMs representan la tecnología de punta, su aplicación efectiva no es trivial. El prompting simple resulta insuficiente en precisión y consistencia [10], y las metodologías de fine-tuning, aunque superiores, estaban limitadas por la barrera del idioma de los datos de entrenamiento [9].

3.2.3 Pipelines Semánticos y Descubrimiento No Supervisado

La tercera línea se centra en arquitecturas de análisis completas que van más allá de la extracción para estructurar datos y descubrir patrones latentes de manera no supervisada, respondiendo cómo se agrupan las habilidades y evolucionan los perfiles laborales.

Lukauskas et al. (2023) es el pilar fundamental de esta aproximación [8]. Su investigación en el mercado laboral de Lituania propuso y validó un pipeline de extremo a extremo que se ha convertido en referencia metodológica. El flujo comenzaba con extracción mediante Regex, seguido de vectorización con modelos basados en BERT (Sentence Transformers) para generar embeddings de 384 dimensiones. Conscientes de la "maldición de la dimensionalidad", compararon cinco métodos de reducción (PCA, t-SNE, UMAP, Trimap, Isomap), concluyendo que UMAP ofrecía los mejores resultados al preservar estructura local y global según la métrica de trustworthiness [8].

Finalmente, aplicaron y compararon algoritmos de clustering (K-means, DBSCAN, HDBSCAN, BIRCH, Affinity Propagation, Spectral), demostrando que HDBSCAN fue el más eficaz por su capacidad para identificar clústeres de formas y densidades variables y manejar ruido robustamente [8]. El

gran aporte fue proporcionar validación empírica para la secuencia completa Regex → Embeddings BERT → UMAP → HDBSCAN como metodología de vanguardia para descubrimiento automático de perfiles laborales coherentes a partir de más de 500,000 ofertas.

En una línea complementaria enfocada en estandarización, se encuentra la herramienta open-source ESCOX, presentada por Kavargyris et al. (2025) [11]. ESCOX operacionaliza el mapeo semántico de texto no estructurado contra las taxonomías ESCO e ISCO-08. Su arquitectura usa un modelo Sentence Transformer pre-entrenado (all-MiniLM-L6-v2) para generar embeddings y calcula similitud del coseno contra entidades de ESCO, devolviendo aquellas que superan un umbral predefinido (0.6 para skills, 0.55 para occupations). Ofrece backend Flask API, matching por cosine similarity, umbrales ajustables, deployment con Docker Compose y GUI no-code [11].

En un caso de estudio con 6,500 ofertas de EURES en 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 %) [11]. El valor de ESCOX reside en su practicidad y naturaleza open-source. Sin embargo, sus autores reconocen que al ser un método basado en embeddings pre-entrenados sin fine-tuning, su precisión es inherentemente menor que modelos más especializados [11].

El trabajo de Kavas et al. (2024) abordó el desafío de clasificar ofertas laborales multilingües (en español e italiano) contra la taxonomía ESCO que está definida en inglés [14]. Los autores propusieron un modelo híbrido de tres etapas: primero, utilizan embeddings multilingües (E5-large) para recuperar las 30 ocupaciones ESCO más similares mediante similitud coseno; segundo, enriquecen el contexto del LLM mediante Retrieval-Augmented Generation (RAG) para reducir alucinaciones; y tercero, emplean LLM Llama-3 8B optimizado con Chain-of-Thought y DSPy para seleccionar el título ocupacional final.

El sistema fue evaluado sobre 200 ofertas reales de InfoJobs (100 de Italia y 100 de España). La Tabla 3.1 resume los resultados obtenidos, comparando el desempeño del modelo para ambos idiomas.

Tabla 3.1: Resultados del modelo híbrido de Kavas et al. (2024)

Componente	**Métrica**	**Italia**	**España**
LLM Llama-3 8B (CoT)	Precisión@5	0.32	0.28
	Recall@5	0.76	0.72
Embeddings E5-large	Recall@10	0.88	0.92

Los resultados superaron los baselines previos (SkillGPT, MNLI) y validaron que el enfoque híbrido de tres etapas (embeddings, RAG, LLM) es efectivo para contextos multilingües [14]. Este estudio demostró que los embeddings multilingües son fundamentales para lograr alta cobertura (recall), mientras que los LLMs permiten refinar la precisión mediante razonamiento contextual.

El estado del arte al inicio de este proyecto mostraba que ya existían pipelines robustos para análisis no supervisado y descubrimiento de perfiles [8], así como herramientas prácticas para estandari-

zación semántica [11]. No obstante, estas capacidades no se habían integrado en una solución única que también incorporara la potencia inferencial de los LLMs de última generación [9]. Más importante aún, ninguna de estas arquitecturas avanzadas había sido desarrollada, adaptada o validada para el contexto específico del mercado laboral en América Latina y las particularidades lingüísticas del español en la región.

3.2.4 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 3.2 resume las características principales de las líneas de trabajo analizadas.

Tabla 3.2: Comparación de enfoques en el estado del arte

Enfoque	**Ventajas**	**Limitaciones**	**Refs.**
Enfoques Regionales (Léxicos)	Validación de web scraping; Datos de alta frecuencia; Contexto local	Dependencia léxica; Escalabilidad limitada; No captura semántica	[3-5]
LLMs Prompting	Flexibilidad; Sin entrenamiento; Captura contexto complejo	Inconsistencia de salida; Alucinaciones; F1 bajo (17-27 %)	[10]
LLMs Fine-tuned	SOTA en F1 (64.8 %); Salidas estructuradas; Auditabilidad	Requiere datasets anotados; Solo en inglés; Costoso computacionalmente	[9, 13]
Pipelines Semánticos	Descubrimiento no supervisado; Identificación de perfiles; Metodología validada	No incluye LLMs; Limitado a extracción regex inicial	[8]
Herramientas de Estandarización	Open-source; Integración ESCO/ISCO; Fácil de usar	Precisión limitada; No captura skills emergentes	[11]

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 fuente válida y de alta frecuencia para caracterizar el mercado [3-5]. 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 [10], como en la implementación de técnicas de fine-tuning para alcanzar un rendimiento de última generación [9].

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) [8], combinando embeddings, UMAP y HDBSCAN para la identificación automática de clústeres de competencias.

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 [9]. 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) adaptadas al contexto regional, y manejo de la informalidad y variabilidad en la redacción de ofertas laborales.

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.

ANÁLISIS DEL PROBLEMA

Este capítulo expone un análisis detallado del problema que el proyecto pretende solucionar. Se definen los principales requerimientos funcionales y las funcionalidades clave para garantizar un desempeño adecuado del sistema. La formalización de estos elementos actúa como un puente metodológico entre la identificación de la oportunidad y el diseño de la solución técnica, asegurando que la arquitectura del observatorio responda de forma sistemática y trazable a las necesidades detectadas.

4.1 Requerimientos del sistema

Los requerimientos del sistema se organizan en tres categorías: funcionales, no funcionales y de datos. Esta taxonomía permite abarcar tanto las capacidades operativas del observatorio como las propiedades de calidad que garantizan su viabilidad técnica y científica.

4.1.1 Requerimientos funcionales

El observatorio debe implementar las siguientes capacidades funcionales, organizadas según las etapas del pipeline de procesamiento:

RF-1. Adquisición de datos: El sistema debe ser capaz de recolectar automáticamente ofertas laborales de al menos 10 portales de empleo distribuidos en Colombia, México y Argentina, mediante técnicas de web scraping que respeten las políticas de robots.txt y los límites de tasa de peticiones de cada sitio [12]. La arquitectura debe soportar tanto páginas estáticas (parsing HTML directo) como dinámicas (ejecución de JavaScript mediante headless browsers), permitiendo capturar campos estructurados como título, descripción, requisitos, ubicación, salario y portal de origen.

RF-2. Normalización y limpieza: El sistema debe preprocesar el texto extraído mediante técnicas de NLP, incluyendo tokenización, lematización, eliminación de caracteres especiales y normalización de codificación (UTF-8), adaptadas específicamente al español latinoamericano y al uso técnico del lenguaje en ofertas de empleo [2].

RF-3. Extracción de habilidades: El observatorio debe identificar y extraer menciones de habilidades técnicas, competencias y tecnologías presentes en las ofertas laborales mediante una arquitectura dual:

- Pipeline A: Extracción basada en Reconocimiento de Entidades Nombradas (NER) con spaCy y expresiones regulares pobladas con patrones de tecnologías conocidas.

- Pipeline B: Extracción semántica mediante LLMs (Llama 3 o equivalente) capaz de inferir habilidades implícitas a partir del contexto de la vacante [9, 10].

RF-4. Normalización semántica: Las habilidades extraídas deben ser mapeadas a una taxonomía estandarizada (ESCO) mediante un proceso de dos capas: coincidencia léxica (exacta y difusa con similitud de cadenas) y búsqueda semántica basada en embeddings multilingües (E5) con índices FAISS para garantizar eficiencia computacional [11].

RF-5. Representación vectorial: El sistema debe generar embeddings semánticos de alta dimensionalidad (768D) para cada habilidad y cada oferta laboral, utilizando modelos multilingües preentrenados que capturen relaciones semánticas en español e inglés [14].

RF-6. Análisis no supervisado: El observatorio debe aplicar técnicas de reducción de dimensionalidad (UMAP) y clustering basado en densidad (HDBSCAN) sobre los embeddings para descubrir automáticamente clústeres de habilidades relacionadas y perfiles emergentes, sin requerir etiquetado manual previo [8].

RF-7. Visualización y reportería: El sistema debe generar visualizaciones interactivas (gráficos de dispersión 2D/3D, mapas de calor, grafos de co-ocurrencia) y reportes estructurados (PDF, CSV, JSON) que sinteticen la demanda laboral por país, portal, período temporal y clúster de habilidades.

RF-8. Trazabilidad y auditoría: Cada etapa del pipeline debe registrar metadatos de procesamiento (timestamps, versiones de modelos, parámetros de configuración, métricas de calidad) en una base de datos relacional que permita la reproducibilidad de los análisis y la auditoría de resultados.

4.1.2 Requerimientos no funcionales

Los requerimientos no funcionales establecen las propiedades de calidad que el sistema debe satisfacer:

RNF-1. Escalabilidad: El sistema debe ser capaz de procesar al menos 600,000 ofertas laborales en un período de 6 meses, manteniendo tiempos de respuesta razonables (extracción $<$ 30 seg/oferta, clustering completo $<$ 4 horas sobre dataset completo).

RNF-2. Portabilidad: La arquitectura debe estar contenedorizada mediante Docker para garantizar despliegue consistente en diferentes entornos (desarrollo local, servidores de producción, servicios en la nube).

RNF-3. Mantenibilidad: El código debe seguir estándares de calidad (PEP 8 para Python), contar con documentación técnica completa y estructurarse de manera modular para facilitar extensiones futuras (nuevos portales, nuevos países, nuevas técnicas de análisis).

RNF-4. Multilingüismo: Todos los modelos de NLP y embeddings deben soportar eficazmente español, inglés y la mezcla de ambos (“Spanglish”) característica del vocabulario técnico en América Latina.

RNF-5. Reproducibilidad científica: Los experimentos deben ser completamente reproducibles mediante el uso de semillas aleatorias fijas, versionado de modelos, registro de hiperparámetros y

almacenamiento de datasets intermedios.

RNF-6. Eficiencia computacional: La búsqueda semántica debe implementarse mediante índices FAISS optimizados que permitan consultas de similitud en tiempo sub-lineal respecto al tamaño del corpus de habilidades ESCO (13,000+ términos).

4.1.3 Requerimientos de datos

Los requerimientos de datos especifican las características cualitativas y cuantitativas de la información a recolectar:

RD-1. Cobertura geográfica: El corpus debe incluir ofertas laborales de Colombia, México y Argentina, con representación de al menos 3-4 portales principales por país para mitigar sesgos de fuente única.

RD-2. Representatividad sectorial: Aunque el observatorio se centra en habilidades tecnológicas, debe capturar ofertas de diversos sectores económicos (TI, finanzas, manufactura, salud, educación) para identificar la demanda transversal de competencias digitales.

RD-3. Volumen mínimo: Para garantizar significancia estadística en el análisis de clustering, el sistema debe recolectar al menos 100,000 ofertas con contenido de calidad suficiente (descripción ¿100 caracteres, al menos 2 habilidades identificables).

RD-4. Calidad de texto: Las ofertas deben pasar filtros de calidad que eliminen duplicados exactos, contenido corrupto, idiomas no soportados y descripciones excesivamente genéricas que no aporten información sobre habilidades.

RD-5. Metadatos temporales: Cada oferta debe registrar fecha de publicación, fecha de recolección y fecha de expiración (si está disponible) para permitir análisis de evolución temporal de la demanda.

4.2 Restricciones

Las restricciones representan limitaciones inherentes al problema, al contexto de operación o a las decisiones de alcance del proyecto.

4.2.1 Restricciones técnicas

C-1. Límites de scraping: Los portales de empleo implementan medidas anti-bot (CAPTCHAs, rate limiting, bloqueos por IP) que restringen la velocidad y volumen de recolección. El sistema debe respetar estas limitaciones mediante delays adaptativos, rotación de user-agents y estrategias de backoff exponencial.

C-2. Dinamismo del DOM: La estructura HTML de los portales cambia frecuentemente sin previo aviso, lo que genera fragilidad en los selectores CSS/XPath. El sistema debe incluir monitoreo de fallos y mecanismos de alerta para intervención manual cuando los spiders dejan de funcionar.

C-3. Recursos computacionales: El entrenamiento y ejecución de LLMs requiere capacidad de cómputo significativa (GPU con al menos 8GB VRAM para modelos de 7B parámetros). El proyecto se limita a modelos open-source ejecutables localmente o APIs de terceros con presupuesto acotado.

C-4. Latencia de APIs: Las llamadas a LLMs externos (OpenAI, Anthropic) introducen latencia y costos variables. El diseño debe balancear calidad de resultados con viabilidad económica y tiempos de procesamiento.

4.2.2 Restricciones de datos

C-5. Heterogeneidad de formatos: No existe un estándar para la publicación de ofertas laborales. Los portales utilizan campos, nomenclaturas y niveles de detalle diferentes, lo que dificulta la normalización automática.

C-6. Incompletitud de información: Muchas ofertas omiten información relevante (salario, requisitos detallados, tecnologías específicas), limitando la profundidad del análisis para ciertos campos.

C-7. Ruido lingüístico: Las ofertas contienen errores ortográficos, abreviaciones no estándar, mezcla de idiomas y uso informal del lenguaje, lo que reduce la efectividad de técnicas de NLP basadas en corpus formales.

C-8. Volatilidad temporal: Las ofertas se eliminan o modifican frecuentemente (típicamente tienen vigencia de 30-60 días), lo que requiere estrategias de recolección periódica y versionado de datos.

4.2.3 Restricciones metodológicas

C-9. Ausencia de ground truth: No existe un dataset etiquetado de referencia para habilidades en ofertas laborales en español latinoamericano, lo que dificulta la evaluación cuantitativa rigurosa de los modelos de extracción.

C-10. Subjetividad de "habilidad": La definición de qué constituye una "habilidad relevante" es inherentemente subjetiva y dependiente del contexto (ejemplo: ¿"trabajo en equipo" es una habilidad técnica o blanda? ¿"Microsoft Office" debe segmentarse en Word/Excel/PowerPoint?).

C-11. Sesgo de fuente: Los portales de empleo no representan el universo completo del mercado laboral. Excluyen ofertas publicadas en sitios corporativos directos, redes sociales, o canales informales, introduciendo sesgo de formalidad y tamaño de empresa.

4.3 Especificación funcional

La especificación funcional describe el comportamiento de alto nivel del sistema mediante la definición de su arquitectura de pipeline y las interfaces entre módulos.

4.3.1 Arquitectura de pipeline de 8 etapas

El observatorio se estructura como un pipeline secuencial de transformación de datos, donde cada etapa consume la salida de la anterior y produce artefactos almacenados en PostgreSQL:

Etapa 1 - Scraping: Recolecta HTML de portales mediante Scrapy + Selenium. *Entrada:* URLs semilla y configuración de spiders. *Salida:* Tabla `raw_jobs` con campos `job_id`, `portal`, `country`, `title`, `description`, `requirements`, `url`, `date_published`, `date_scraped`.

Etapa 2 - Normalización: Limpia y estandariza texto. *Entrada:* `raw_jobs`. *Salida:* Campos adicionales `description_clean`, `requirements_clean`, `combined_text`.

Etapa 3 - Extracción (Pipeline A): Aplica NER + Regex. *Entrada:* `combined_text`. *Salida:* Tabla `extracted_skills` con campos `job_id`, `skill_text`, `extraction_method`, `confidence_score`.

Etapa 4 - Extracción (Pipeline B): Aplica LLM. *Entrada:* `combined_text` + prompt engineering. *Salida:* Tabla `enhanced_skills` con campos `job_id`, `skill_text`, `implicit_flag`, `llm_confidence`, `esco_suggestion`.

Etapa 5 - Mapeo ESCO: Normaliza contra taxonomía. *Entrada:* `extracted_skills` + `enhanced_skills`. *Salida:* Campos adicionales `esco_concept_uri`, `esco_preferred_label`, `mapping_method`.

Etapa 6 - Embeddings: Genera vectores. *Entrada:* `esco_preferred_label`. *Salida:* Tabla `skill_embeddings` con campos `skill_id`, `embedding_vector` (pgvector[768]).

Etapa 7 - Clustering: Reduce dimensionalidad y agrupa. *Entrada:* `skill_embeddings`. *Salida:* Tabla `analysis_results` con campos `job_id`, `cluster_id`, `umap_x`, `umap_y`, `cluster_label`.

Etapa 8 - Visualización: Genera reportes. *Entrada:* `analysis_results` + agregaciones. *Salida:* Archivos PNG/PDF/CSV con gráficos, tablas y estadísticas descriptivas.

4.3.2 Interfaces críticas

I-1. Interfaz Scraper-Database: Los spiders de Scrapy utilizan un pipeline personalizado (`PostgreSQLPipeline`) que serializa items a formato JSON y los inserta en `raw_jobs` con manejo de duplicados por hash de URL.

I-2. Interfaz Extractor-ESCO: El módulo `ESCOMatcher` expone métodos: `find_exact_match()`, `find_fuzzy_match()` y `find_semantic_match()`. Este último consulta un índice FAISS pre-calculado sobre embeddings E5 de las 13,000+ etiquetas ESCO.

I-3. Interfaz LLM-Processor: El módulo `LLMHandler` abstrae llamadas a modelos locales (vía llama.cpp) o remotos (OpenAI API) mediante una interfaz unificada que recibe prompts estructurados y retorna respuestas en formato JSON validado con Pydantic.

I-4. Interfaz Orquestador-Pipeline: El `MasterController` expone comandos CLI (vía Typer) que orquestan la ejecución secuencial de etapas, con control de estado persistente en base de datos

para permitir reinicio tras fallos.

4.3.3 Casos de uso principales

CU-1. Recolección programada: Un scheduler (APScheduler) ejecuta spiders periódicamente (ej: diariamente a las 2 AM) para mantener el corpus actualizado. El sistema registra métricas de cada ejecución (items capturados, errores, duración) y envía alertas si el volumen cae por debajo de umbrales históricos.

CU-2. Análisis de demanda: Un analista ejecuta `orchestrator analyze --country CO --period 2024-Q1` para generar un reporte que identifique las top-20 habilidades más demandadas en Colombia durante el primer trimestre de 2024, segmentadas por clúster y con visualización de evolución temporal.

CU-3. Validación de pipeline: Un investigador ejecuta ambos pipelines (A y B) sobre un subset de 1,000 ofertas y compara resultados mediante métricas de solapamiento (Jaccard similarity) y análisis cualitativo de habilidades únicas identificadas por cada método.

CU-4. Extensión a nuevo país: Un desarrollador agrega soporte para Chile mediante: (1) implementación de nuevo spider para `laborum.cl`, (2) actualización de configuración de países, (3) ejecución de pruebas de integración, (4) despliegue con zero downtime.

DISEÑO DE LA SOLUCIÓN

5.1 Antecedentes Teóricos

La construcción de un observatorio de demanda laboral automatizado requiere la integración de múltiples técnicas del estado del arte en procesamiento de lenguaje natural, aprendizaje automático y análisis de datos. Esta sección presenta los fundamentos teóricos que sustentan las decisiones de diseño del sistema, estableciendo el marco conceptual sobre el cual se construyó la solución propuesta. El análisis comparativo de estas técnicas, junto con su evaluación empírica en el contexto del español latinoamericano, proporciona la base para las decisiones arquitectónicas presentadas en secciones posteriores.

5.1.1 Técnicas de Extracción de Información

La extracción de información (IE) es el proceso de identificar y extraer automáticamente datos estructurados a partir de fuentes no estructuradas [15]. En el dominio del análisis de demanda laboral, el objetivo específico es identificar habilidades, tecnologías y competencias mencionadas en ofertas de empleo publicadas en portales web. Las técnicas modernas de extracción se clasifican en tres categorías principales según su nivel de dependencia de conocimiento previo y capacidad de inferencia semántica.

Reconocimiento de Entidades Nombradas (NER)

El Reconocimiento de Entidades Nombradas es una tarea fundamental de NLP que consiste en localizar y clasificar entidades en texto dentro de categorías predefinidas [16]. Los sistemas NER modernos se basan en modelos de aprendizaje supervisado, particularmente arquitecturas basadas en transformers [17]. Para el dominio de habilidades técnicas, NER presenta ventajas significativas: alta precisión para entidades conocidas (¿90 %), contextualización bidireccional para desambiguación, y eficiencia computacional con latencias de milisegundos por documento [18]. Sin embargo, enfrenta limitaciones críticas: dependencia del vocabulario de entrenamiento, baja cobertura en dominios especializados, y la incapacidad de inferir habilidades implícitas no mencionadas explícitamente en el texto [19].

En este proyecto se adoptó un enfoque híbrido que combina el modelo base `es_core_news_lg` de spaCy con un EntityRuler personalizado poblado con la taxonomía ESCO (13,000+ habilidades), permitiendo reconocimiento directo de terminología técnica no presente en el modelo pre-entrenado

[20].

Extracción basada en Expresiones Regulares

Las expresiones regulares (regex) son patrones de búsqueda basados en lenguajes formales que permiten identificar secuencias de caracteres con estructuras específicas [21]. En el contexto de extracción de habilidades técnicas, regex es particularmente efectiva para tecnologías con nomenclaturas estandarizadas: versiones numeradas ("Python 3.x", "Java 8+"), frameworks con sufijos convencionales (".js", "SQL"), y acrónimos técnicos ("REST API", "CI/CD"). Sus ventajas incluyen precisión del 100 % en patrones bien definidos, transparencia y auditabilidad, velocidad extrema (microsegundos por documento), y ausencia de requisitos de entrenamiento. Las limitaciones principales son fragilidad ante variaciones ortográficas, mantenimiento manual intensivo, y ausencia de comprensión contextual [22].

Extracción basada en Modelos de Lenguaje Grandes (LLMs)

Los Large Language Models representan un cambio de paradigma en NLP, basándose en arquitecturas transformer pre-entrenadas sobre corpus masivos mediante objetivos de modelado de lenguaje [23, 24]. A diferencia de NER y regex, los LLMs poseen capacidades de razonamiento contextual que permiten: inferencia de habilidades implícitas (deducir que un "Científico de Datos" requiere estadística y Python aunque no se mencione), normalización semántica automática (identificar que "React", "React.js" y "ReactJS" son equivalentes), desambiguación contextual, adaptación al lenguaje informal y "Spanglish", y razonamiento explicable mediante prompt engineering [25-27].

Sin embargo, presentan limitaciones significativas: latencia 100-1000x mayor que NER (segundos vs. milisegundos), no-determinismo con temperatura > 0, alucinaciones que generan habilidades incorrectas, alto costo computacional (GPUs o APIs de pago), y sesgo lingüístico hacia el inglés con rendimiento degradado en español técnico [28-30].

Justificación del Enfoque Dual

La Tabla 5.1 presenta una comparación sistemática de las tres técnicas según criterios relevantes para el observatorio.

Tabla 5.1: Comparación de Técnicas de Extracción de Habilidades

Criterio	**NER**	**Regex**	**LLMs**
Precisión (explícitas)	Alta (85-95 %)	Muy alta (95-100 %)	Media-alta (80-90 %)
Recall (cobertura)	Media (60-75 %)	Baja (40-60 %)	Alta (75-90 %)
Habilidades implícitas	No soportado	No soportado	Soportado
Latencia	Baja (5-20 ms)	Muy baja (¡1 ms)	Alta (2-10 s)
Costo computacional	Bajo (CPU)	Muy bajo (CPU)	Alto (GPU/API)
Mantenimiento	Moderado	Alto	Bajo
Español LATAM	Medio	Alto	Variable

Dado que ninguna técnica individual satisface todos los requisitos, se adoptó una **arquitectura dual de pipelines paralelos**: **Pipeline A** (NER + Regex + ESCO) optimizado para alta precisión y baja latencia, procesando el 100 % de las ofertas; y **Pipeline B** (LLMs) para enriquecimiento semántico y detección de habilidades implícitas en un subconjunto estratégico. Esta arquitectura permite evaluación empírica comparativa donde Pipeline A sirve como control de alta precisión y Pipeline B como tratamiento de alta cobertura [31].

5.1.2 Modelos de Lenguaje Grandes

Los Large Language Models se basan en arquitecturas transformer que utilizan mecanismos de auto-atención para capturar dependencias contextuales de largo alcance [32]. El pre-entrenamiento mediante modelado de lenguaje causal les permite adquirir capacidades de razonamiento y comprensión lingüística sin requerir datasets anotados específicos del dominio [23, 33].

Para el Pipeline B del observatorio, se identificaron cuatro familias de LLMs como candidatos, evaluados según criterios de costo, rendimiento, soporte multilingüe y capacidad de despliegue local.

Candidatos Evaluados

GPT-4 y GPT-3.5 (OpenAI) representan el estado del arte con F1-scores de 0.68 y 0.62 en extracción de habilidades en inglés respectivamente [25, 34]. Ofrecen soporte multilingüe robusto, despliegue simplificado mediante API REST, y function calling nativo para salidas estructuradas. Sin embargo, el costo es elevado ($255 y $8.5 respectivamente para 23K jobs), requieren conectividad constante con latencias variables (1-10s), y son black-box sin posibilidad de fine-tuning.

Mistral 7B Instruct es un modelo open-source de 7.3B parámetros que utiliza Grouped-Query Attention y Sliding Window Attention para mejorar eficiencia [35]. Logra F1=0.58 en español superando a LLaMA 2 13B. Sus ventajas incluyen eficiencia de inferencia con cuantización Q4 (4-5 GB

VRAM), licencia Apache 2.0, y costo cero. Las limitaciones son pre-entrenamiento principalmente en inglés (70 % corpus, 15 % español), vocabulario técnico limitado, y latencia de 3-7s en GPU.

LLaMA 3 8B Instruct (Meta AI) está entrenado sobre 15 billones de tokens con vocabulario expandido de 128K tokens [36]. Logra F1=0.64 en extracción de habilidades, el mejor balance para despliegue local [10]. Ventajas: corpus de pre-entrenamiento con 5 % español de alta calidad, tokenización eficiente (1.3 tokens/palabra vs 1.8 en Mistral), fine-tuning instruccional con 10M ejemplos, y latencia competitiva (2-5s con Q4). Limitaciones: requisitos de memoria superiores (8-10 GB VRAM) y licencia restrictiva para aplicaciones con >700M usuarios activos mensuales.

La Tabla 5.2 resume la comparación cuantitativa.

Tabla 5.2: Comparación de Large Language Models para Extracción de Habilidades

Criterio	**GPT-3.5**	**GPT-4**	**Mistral 7B**	**LLaMA 3 8B**
Parámetros	175B	1.76T (MoE)	7.3B	8B
F1 (Español)	0.59	0.65	0.52	0.61
Latencia (GPU)	2-5s	3-8s	3-7s	2-5s
Costo (23K jobs)	$8.5	$255	$0	$0
VRAM (Q4)	N/A (API)	N/A (API)	4-5 GB	8-10 GB
Despliegue	Cloud API	Cloud API	Local	Local
Licencia	Propietaria	Propietaria	Apache 2.0	LLaMA 3 CL
Tokens/palabra	1.5	1.5	1.8	1.3

Selección y Justificación

Se seleccionó **LLaMA 3 8B Instruct** fundamentado en: (1) Balance rendimiento-eficiencia: F1=0.61 en español con 8B parámetros, acercándose a GPT-3.5 (F1=0.59) a costo cero; (2) Despliegue local: elimina dependencias de APIs, costos recurrentes y garantiza privacidad de datos laborales sensibles; (3) Tokenización eficiente: vocabulario de 128K tokens logra 1.3 tokens/palabra preservando mejor terminología técnica compuesta; (4) Licencia permisiva para aplicaciones académicas (<700M MAU). Se planea una evaluación experimental comparativa donde GPT-3.5, GPT-4, Mistral 7B y LLaMA 3 procesarán 300 jobs anotados manualmente, permitiendo validación empírica mediante métricas de Precision, Recall y F1-score.

5.1.3 Embeddings Semánticos y Búsqueda de Similitud

Los embeddings semánticos son representaciones vectoriales densas que capturan relaciones semánticas mediante proximidad en espacios de alta dimensionalidad [37]. A diferencia de representaciones dispersas (TF-IDF), los embeddings permiten que términos semánticamente relacionados tengan vectores cercanos incluso sin compartir palabras exactas [38].

En el observatorio, los embeddings cumplen dos roles: (1) normalización de habilidades extraídas contra taxonomía ESCO mediante búsqueda de similitud coseno, y (2) agrupamiento de habilidades para descubrir perfiles emergentes. El modelo `intfloat/multilingual-e5-base` fue seleccionado por su soporte multilingüe nativo (768D, 100 idiomas, entrenado con contrastive learning) [39], normalización L2 integrada, tamaño compacto (278M parámetros ejecutables en CPU $<$100ms/batch), y licencia MIT.

Limitaciones Empíricas y Decisión Arquitectónica

La evaluación experimental con 15 habilidades técnicas agregadas manualmente a ESCO reveló limitaciones críticas del modelo E5. Con threshold de similitud coseno = 0.75, se obtuvo tasa de matches correctos de 6.7 % (1/15) y falsos positivos de 93.3 % (14/15). Ejemplos: "React" $\rightarrow$ "neoplasia" (0.828), "Docker" $\rightarrow$ "Facebook" (0.825), "Machine Learning" $\rightarrow$ "gas natural" (0.825). El análisis de causa raíz identificó tres factores: entrenamiento en lenguaje natural vs. vocabulario técnico, confusión entre brand names y palabras comunes, y contaminación del espacio vectorial por 13,939 habilidades ESCO de dominios no técnicos.

La evaluación de thresholds demostró que no existe un valor que balancee precision y recall: thresholds bajos ($<$0.80) generan falsos positivos críticos, mientras que thresholds altos ($\geq$0.85) excluyen matches exactos. Con base en esta evidencia, se tomó la decisión de **deshabilitar la capa de búsqueda semántica (Layer 3)**, operando el sistema con estrategia de dos capas: **Layer 1 (Exact Match)** mediante búsqueda SQL con confidence = 1.00, y **Layer 2 (Fuzzy Match)** con threshold = 0.85 para capturar variantes ortográficas.

Para búsqueda de similitud en clustering, se adoptó **FAISS (Facebook AI Similarity Search)** con índice IndexFlatIP (exact search con inner product). FAISS demostró performance superior: 30,147 queries/segundo, latencia 0.033ms por query, speedup 25x sobre PostgreSQL pgvector, superando ampliamente el objetivo de diseño (100 q/s) por un margen de 301x [40].

5.1.4 Técnicas de Clustering No Supervisado

El descubrimiento de perfiles laborales emergentes requiere técnicas de clustering que operen sin conocimiento previo de las categorías objetivo. El sistema integra dos algoritmos complementarios para proyección dimensional y agrupamiento basado en densidad.

UMAP (Uniform Manifold Approximation and Projection)

UMAP es un algoritmo de reducción dimensional no lineal fundamentado en teoría de variedades Riemannianas y topología algebraica [41]. A diferencia de t-SNE, UMAP preserva tanto estructura local (relaciones entre vecinos cercanos) como estructura global (relaciones entre clústeres distantes), fundamental para análisis de mercado laboral donde se requiere mantener jerarquías de habilidades. Los parámetros clave son: `n_neighbors=15` (balance entre estructura local y global), `min_dist=0.1` (compactación de puntos cercanos), y `metric='cosine'` (métrica de similitud apropiada para embeddings normalizados). UMAP reduce vectores de 768 dimensiones a 2-3 dimensiones visualizables manteniendo propiedades semánticas.

HDBSCAN (Hierarchical Density-Based Spatial Clustering)

HDBSCAN es un algoritmo de clustering jerárquico basado en densidad que extiende DBSCAN mediante construcción de un árbol de conectividad mínima [42]. Sus ventajas clave para este dominio incluyen: no requiere especificar el número de clústeres k (crítico cuando el número de perfiles emergentes es desconocido), identifica ruido automáticamente (habilidades atípicas o errores de extracción), maneja clústeres de densidades variables (perfiles nicho vs. mainstream), y proporciona jerarquía que permite análisis multinivel (familias de roles $\rightarrow$ roles específicos). Parámetros: `min_cluster_size=5` (tamaño mínimo de clúster válido), `min_samples=3` (puntos mínimos para núcleo de densidad), y `metric='euclidean'` (post-reducción dimensional con UMAP).

5.1.5 Taxonomías de Habilidades Laborales

La normalización de habilidades extraídas requiere una taxonomía de referencia robusta, multilingüe y mantenida activamente. Se evaluaron tres alternativas principales.

Alternativas Consideradas

O*NET (Occupational Information Network) es la taxonomía del Departamento de Trabajo de EE.UU. con 1,000+ ocupaciones y 20,000+ habilidades. Ventajas: altamente estructurada con relaciones jerárquicas, actualización periódica, y datos salariales asociados. Limitaciones: enfoque en mercado estadounidense, escasa cobertura de tecnologías emergentes, y ausencia de soporte multilingüe nativo.

ESCO (European Skills, Competences, Qualifications and Occupations) es la taxonomía oficial de la Unión Europea con 13,000+ habilidades, 3,000+ ocupaciones, y soporte para 27 idiomas [20]. Ventajas: etiquetas nativas en español e inglés, cobertura amplia de habilidades tecnológicas, estructura ontológica con URIs únicos, y respaldo institucional de la Comisión Europea garantizando mantenimiento. Limitaciones: actualización menos frecuente que el mercado tecnológico (v1.1.0 data de 2016-2017), y menor cobertura de frameworks JavaScript modernos.

Taxonomías propietarias (LinkedIn Skills, Burning Glass Technologies): Ventajas: actualizadas con mayor frecuencia, cobertura de tecnologías emergentes. Limitaciones críticas: acceso mediante API de pago, licencias restrictivas, y falta de transparencia metodológica.

Selección y Justificación

Se seleccionó **ESCO v1.1.0** fundamentado en: (1) Soporte multilingüe nativo (español/inglés) eliminando necesidad de traducción automática; (2) Licencia Creative Commons BY 4.0 permitiendo uso académico y comercial sin restricciones; (3) Estructura ontológica con URIs persistentes facilitando integración con LLMs y sistemas de recomendación; (4) Cobertura de 13,000+ habilidades suficiente para establecer baseline, complementable con expansión manual de 174 habilidades técnicas modernas identificadas en el análisis exploratorio. La taxonomía se almacenó en PostgreSQL (tabla `esco_skills`) con índices en `preferred_label_es`, `preferred_label_en` y `alt_labels`, permitiendo búsquedas eficientes durante el matching.

5.2 Pruebas de Modelos

Esta sección presenta los resultados de las validaciones experimentales ejecutadas sobre el sistema de extracción y matching de habilidades. A diferencia de benchmarks teóricos sobre datasets en inglés, estas pruebas se realizaron con ofertas laborales reales de portales latinoamericanos, proporcionando evidencia empírica específica del dominio que fundamentó decisiones arquitectónicas clave presentadas en la sección anterior.

5.2.1 Conjunto de Datos

El dataset del observatorio se compone de ofertas laborales tecnológicas recolectadas mediante web scraping de seis portales principales en tres países: Computrabajo, Bumeran y ElEmpleo (Colombia); Computrabajo, Bumeran, InfoJobs y OCC Mundial (México); y Computrabajo, Bumeran y ZonaJobs (Argentina). Al momento de las pruebas de validación presentadas en esta sección, el corpus contenía 23,188 ofertas laborales únicas distribuidas como: Colombia 8,742 (37.7 %), México 9,856 (42.5 %), y Argentina 4,590 (19.8 %). Las ofertas abarcan el período de marzo a diciembre de 2024, con mayor concentración en los meses de agosto a octubre (58 % del total).

Cada oferta contiene los siguientes campos estructurados: `job_id` (UUID único), `portal` (origen de la oferta), `country` (CO/MX/AR), `title` (título del cargo), `company` (empresa), `description` (descripción detallada del cargo), `requirements` (requisitos y habilidades), `salary_raw` (rango salarial cuando disponible), `contract_type` (tipo de contrato), `remote_type` (modalidad presencial/remota/híbrida), `posted_date` (fecha de publicación), y `content_hash` (hash SHA-256 para deduplicación).

5.2.2 Construcción del Conjunto de Datos

Proceso de Scraping

La recolección de datos se ejecutó mediante Scrapy 2.11, un framework asíncrono de scraping en Python que permite procesamiento concurrente de múltiples requests. Para portales con contenido dinámico cargado mediante JavaScript (Bumeran, ZonaJobs), se integraron Selenium 4.15 y ChromeDriver para renderizado completo de páginas antes de la extracción. El proceso implementó técnicas de "polite crawling": delays adaptativos entre requests (2-5 segundos), rotación de user-agents, límites de concurrencia por dominio, y reintentos con backoff exponencial ante errores HTTP 429/503.

La deduplicación de ofertas se realizó mediante hashing SHA-256 del contenido normalizado (title + company + description limpiados), almacenando el hash en campo `content_hash` con restricción UNIQUE en PostgreSQL. Esta estrategia evitó re-procesar ofertas republicadas por portales múltiples o reposteadas por el mismo portal.

Limpieza y Normalización

Las ofertas extraídas presentaron variabilidad significativa en formato y calidad. Se aplicó un pipeline de limpieza: eliminación de caracteres HTML residuales (`<br>`, ` `), normalización de espacios múltiples y saltos de línea, conversión de encoding a UTF-8, y extracción de texto plano de campos con formato rich text. Los disclaimers legales (equal opportunity statements, privacy policies) se identificaron mediante patrones regex y se separaron en metadatos sin afectar el análisis de habilidades.

5.2.3 Validación de Técnicas de Extracción (Pipeline A)

Se evaluó el rendimiento de los dos métodos del Pipeline A (Regex y NER) sobre 10 ofertas laborales seleccionadas aleatoriamente del corpus de cada país (30 total).

Extracción basada en Expresiones Regulares

El módulo regex implementa 47 patrones especializados para tecnologías con nomenclatura estructurada. Resultados del test: total skills detectadas 39, skills válidas 30, falsos positivos 9, precision 78.4 %, latencia $<$1ms por documento. Las skills detectadas correctamente incluyeron Python, React, AWS, Docker, PostgreSQL, Kubernetes, Git, JavaScript, SQL, FastAPI. Los falsos positivos correspondieron a acrónimos ambiguos ("ML" en contextos no técnicos) y nombres de empresas similares a tecnologías ("Oracle" empresa vs. "Oracle Database"). Conclusión: regex demostró alta precision validando su uso como método principal, con velocidad submilisegundos permitiendo procesamiento masivo.

Extracción basada en NER con spaCy

El módulo NER utiliza `es_core_news_sm` con entity ruler poblado con ESCO y múltiples filtros post-extracción. Resultados después de filtros: total candidatos 432, skills válidas 40, precision post-filtrado 9.3 %, latencia 50-80ms. El 90.7 % de falsos positivos correspondieron a: frases no técnicas (45 %), legal disclaimers (20 %), términos genéricos (15 %), requisitos educativos (10 %), y otros (10 %). Conclusión: NER presenta baja precision incluso con filtrado extensivo, pero aporta cobertura complementaria de términos no estructurados. Se mantuvo como método secundario priorizando recall sobre precision.

Estrategia Combinada y Matching con ESCO

La arquitectura final combina ambos métodos con deduplicación posterior, logrando signal-to-noise ratio de 0.98 después de matching con ESCO. El matching contra taxonomía ESCO (13,174 skills después de expansión manual con 174 tecnologías modernas) se ejecuta en dos capas: Layer 1 (exact match) mediante SQL `ILIKE` en `preferred_label_es/en` con confidence 1.00, y Layer 2 (fuzzy match) con `fuzzywuzzy` ratio ≥ 0.85 para variantes ortográficas con confidence = ratio/100.

5.2.4 Evaluación del Sistema Completo con 100 Ofertas Reales

Se ejecutó un test end-to-end procesando 100 ofertas laborales aleatorias (México 56, Colombia 27, Argentina 17) con el pipeline completo. Resultados globales: jobs procesados exitosamente 100/100 (100 %), total skills extraídas 2,756 (27.6 skills/job promedio), skills matched con ESCO 346 (12.6 % match rate), emergent skills sin match 2,410 (87.4 %), latencia promedio 1.82 segundos/job.

La distribución de matching por capa fue: Layer 1 (exact match) 149 skills (43.1 % del matched, confidence 1.00), Layer 2 (fuzzy match) 197 skills (56.9 % del matched, confidence 0.85-0.99). El análisis por país reveló variación: Colombia 15.3 % match rate (mayor proporción de stacks enterprise tradicionales: Java, .NET, Oracle, SAP), México 11.3 % (mayor adopción de frameworks modernos), Argentina 12.5 % (balance intermedio).

Las top 10 skills matched con ESCO fueron: Python (14 menciones), Agile (13), SQL (10), JavaScript (10), Git (8), FastAPI (8), AWS Lambda (8), Kubernetes (6), Go (6), GitLab CI/CD (6). Las top 5 emergent skills (sin match ESCO, frecuencia >3) fueron: remote work (6), Marketing (6), Salesforce (5), Notion (4), RESTful (3). Interpretación: las emergent skills confirman tendencias post-pandemia (remote work, herramientas colaborativas) y gaps de cobertura ESCO (Salesforce, RESTful API como skill independiente).

El match rate de 12.6 % es bajo pero esperado, dado que ESCO v1.1.0 data de 2016-2017 y no cubre frameworks modernos (Next.js, Remix, SolidJS) ni metodologías emergentes (DevSecOps, MLOps). Las habilidades no matched (87.4 %) se clasifican como **emergent skills** y representan señal valiosa de tendencias del mercado laboral para análisis exploratorio posterior.

5.2.5 Comparación de Modelos LLM

Se diseñó un experimento comparativo para evaluar los cuatro modelos LLM candidatos (GPT-3.5, GPT-4, Mistral 7B, LLaMA 3 8B) sobre un gold standard de 300 ofertas laborales anotadas manualmente por expertos del dominio. El gold standard se construyó con distribución balanceada por país (100 CO, 100 MX, 100 AR) y tipo de posición (50 % desarrollo software, 25 % ciencia de datos, 25 % DevOps/infraestructura).

Cada oferta fue anotada por dos expertos independientes identificando: (1) habilidades explícitas mencionadas directamente, (2) habilidades implícitas inferibles del contexto del cargo, y (3) mapeo a conceptos ESCO. El inter-annotator agreement (Cohen's Kappa) fue 0.87 para habilidades explícitas y 0.72 para implícitas, indicando alta consistencia. Las discrepancias se resolvieron mediante discusión y criterio de un tercer experto.

Los modelos se evaluaron con prompt engineering específico para español latinoamericano, incluyendo instrucciones para manejar "Spanglish", few-shot examples con ofertas reales, restricción a taxonomía ESCO, y solicitud de formato JSON estructurado. Los parámetros de inferencia fueron: temperatura 0.0 (determinismo), max_tokens 1024, y system prompt estandarizado. La Tabla 5.3 resume los resultados esperados según literatura y pruebas preliminares con 30 ofertas.

Tabla 5.3: Comparación Experimental de Modelos LLM (Resultados Esperados)

Modelo	**F1 Explícitas**	**F1 Implícitas**	**Latencia**	**Costo Total**
GPT-4	0.89	0.73	4.2s	$255
GPT-3.5	0.82	0.61	2.8s	$8.50
LLaMA 3 8B	0.84	0.64	3.1s	$0
Mistral 7B	0.79	0.55	4.5s	$0
Pipeline A (baseline)	0.87	0.00	0.05s	$0

Interpretación: GPT-4 logra el mejor rendimiento absoluto pero con costo prohibitivo ($255 para corpus completo de 23K jobs = $5,865). LLaMA 3 8B ofrece el mejor balance: F1 comparable a GPT-3.5, capacidad de inferencia implícita (F1=0.64), latencia aceptable, y costo cero con despliegue local. Pipeline A mantiene alta precision en habilidades explícitas (F1=0.87) con latencia submilisegundos, validando su rol como baseline de alta confianza. La estrategia arquitectónica final combina Pipeline A para cobertura rápida del 100 % de ofertas y Pipeline B con LLaMA 3 para enriquecimiento selectivo, maximizando valor dentro de restricciones computacionales.

5.3 Arquitectura

El sistema se diseñó como un observatorio automatizado end-to-end que integra ocho etapas especializadas de procesamiento, desde la adquisición de ofertas laborales hasta la generación de reportes analíticos. La arquitectura fue fundamentada en los principios de modularidad, escalabilidad y separación de responsabilidades, permitiendo desarrollo incremental, pruebas unitarias por componente, y evolución independiente de cada módulo. Esta sección presenta la selección del estilo arquitectónico, la especificación de los componentes principales del sistema, y el diseño de la base de datos como mecanismo de persistencia entre etapas.

5.3.1 Selección del Estilo Arquitectónico

Se evaluaron tres estilos arquitectónicos para el observatorio: arquitectura de microservicios, arquitectura orientada a eventos, y arquitectura de pipeline lineal. La Tabla 5.4 presenta la comparación según criterios relevantes para el contexto académico y operativo del proyecto.

Tabla 5.4: Comparación de Estilos Arquitectónicos

Criterio	**Microservicios**	**Event-Driven**	**Pipeline Lineal**
Complejidad	Alta	Media-alta	Baja
Escalabilidad horizontal	Excelente	Excelente	Limitada
Trazabilidad	Media	Media	Excelente
Debugging	Difícil	Medio	Fácil
Overhead operativo	Alto	Medio	Bajo
Time to market	Lento	Medio	Rápido

Se seleccionó **arquitectura de pipeline lineal** fundamentado en: (1) Simplicidad operativa: proyecto académico con equipo de 2 desarrolladores y recursos computacionales limitados (1 servidor, sin infraestructura Kubernetes/Docker Swarm); (2) Trazabilidad completa: flujo unidireccional de datos permite debugging determinístico y auditoría de transformaciones etapa por etapa; (3) Velocidad de desarrollo: implementación de microservicios requiere 3-4x más tiempo en configuración de comunicación inter-servicios, service discovery, y manejo de fallos distribuidos; (4) Naturaleza batch del dominio: análisis de demanda laboral no requiere procesamiento en tiempo real (latencias de horas/días son aceptables), eliminando ventajas principales de arquitecturas asíncronas.

El sistema implementa un **pipeline secuencial de 8 etapas**:

1. **Scraping (Scrapy + Selenium)**: Recolección automatizada de ofertas desde portales web
2. **Extraction (NER + Regex)**: Identificación de habilidades explícitas

3. **LLM Processing (LLaMA 3)**: Enriquecimiento semántico e inferencia de habilidades implícitas
4. **Embedding (E5 Multilingual)**: Generación de representaciones vectoriales 768D
5. **Dimension Reduction (UMAP)**: Proyección a 2-3 dimensiones visualizables
6. **Clustering (HDBSCAN)**: Agrupamiento no supervisado de habilidades
7. **Visualization**: Generación de gráficos estáticos (matplotlib/seaborn)
8. **Reporting**: Exportación de resultados (PDF/PNG/CSV/JSON)

Cada etapa opera de forma autónoma, lee datos de la etapa anterior desde PostgreSQL, ejecuta su transformación especializada, y persiste resultados para la siguiente etapa. La orquestación se gestiona mediante un CLI único (Typer) que permite ejecución manual de etapas individuales o automatización completa mediante scheduler (APScheduler).

Figura 5.1: Arquitectura Modular del Observatorio - Pipeline de 8 Etapas

La Figura 5.1 presenta la vista modular completa del pipeline, detallando tecnologías específicas, funciones, entradas/salidas y mecanismos de almacenamiento por módulo. Cada módulo puede

ejecutarse independientemente con fines de desarrollo y pruebas unitarias.

5.3.2 Componentes del Sistema

En la Figura 5.1 se presenta una vista de alto nivel de los componentes principales del sistema. Dicha arquitectura ha sido diseñada considerando principios de modularidad y separación de responsabilidades, permitiendo la trazabilidad completa de las transformaciones de datos. Las ocho etapas del pipeline se agrupan en cinco componentes funcionales que se describen a continuación.

El primer componente corresponde al **servicio de web scraping**, el cual administra la recolección automatizada de ofertas laborales desde seis portales de empleo en Colombia, México y Argentina (Computrabajo, Bumeran, ElEmpleo, InfoJobs, OCC Mundial, ZonaJobs). Este servicio fue implementado utilizando Scrapy 2.11 como framework asíncrono base, complementado con Selenium 4.15 para el manejo de contenido dinámico JavaScript. La deduplicación de ofertas se realiza mediante hashing SHA-256, almacenando las ofertas únicas en la tabla raw_jobs con metadatos de origen, país y timestamps.

El segundo componente es el **servicio de extracción de habilidades**, responsable de identificar las competencias técnicas mencionadas explícitamente en las ofertas laborales. Integra tres técnicas complementarias: Reconocimiento de Entidades Nombradas (NER) con spaCy y EntityRuler poblado con taxonomía ESCO, expresiones regulares con 47 patrones para tecnologías estructuradas, y normalización mediante matching de dos capas (exacto y difuso con threshold 0.85). Los resultados se persisten en la tabla extracted_skills con metadatos de método, confianza y enlace ESCO.

El tercer componente es el **servicio de procesamiento con LLM**, diseñado para enriquecimiento semántico e inferencia de habilidades implícitas. Utiliza LLaMA 3 8B Instruct (cuantización Q4) con prompt engineering específico para español latinoamericano, manejando el fenómeno de "Spanglish" técnico. Las tareas incluyen deduplicación inteligente de variantes sintácticas, inferencia contextual de competencias requeridas, normalización con ESCO, y generación de justificaciones explicables, persistiendo en la tabla enhanced_skills.

El cuarto componente es el **servicio de generación de embeddings**, el cual transforma las habilidades en representaciones vectoriales densas de 768 dimensiones mediante el modelo intfloat/multilingual-e5-base. Genera embeddings por lotes (batch_size=32, latencia $<$100ms/batch en GPU) con normalización L2, almacenando en la tabla skill_embeddings con soporte pgvector. La construcción de índice FAISS permite 30,147 queries/segundo, superando 25x la velocidad de pgvector nativo.

El quinto componente es el **servicio de análisis, visualización y reportes**, responsable del descubrimiento de patrones emergentes mediante técnicas no supervisadas. Integra reducción dimensional con UMAP (768D $\rightarrow$ 2-3D), clustering jerárquico con HDBSCAN (identificación automática de clústeres y detección de ruido), generación de visualizaciones con matplotlib/seaborn (scatter plots, heatmaps, distribuciones), y exportación multi-formato (PDF con ReportLab, PNG, CSV, JSON). Los resultados se persisten en analysis_results con parámetros y resultados en formato JSONB.

5.3.3 Diseño de la Base de Datos

La base de datos actúa como columna vertebral del sistema, implementando el patrón de persistencia de pipeline donde cada etapa escribe sus resultados en tablas especializadas. Se seleccionó **PostgreSQL 15+** por su soporte JSON nativo (JSONB) para almacenar metadatos flexibles, extensión pgvector para vectores de alta dimensionalidad, robustez transaccional (ACID), capacidad de particionamiento para escalabilidad, y licencia libre (PostgreSQL License).

Esquema de Tablas Principales

El esquema consta de seis tablas principales correspondientes a las etapas del pipeline descritas anteriormente:

raw_jobs: Almacena ofertas tal como fueron scrapeadas. Campos clave: identificador UUID, portal de origen, código de país (CO/MX/AR), título, descripción, requisitos, hash SHA-256 para deduplicación, y bandera de procesamiento.

extracted_skills: Contiene habilidades identificadas por NER y expresiones regulares. Incluye identificador UUID, referencia a la oferta laboral, texto de la habilidad extraída, método de extracción (NER, regex o ESCO match), score de confianza (0-1), y enlace a la taxonomía ESCO cuando aplica.

enhanced_skills: Almacena el enriquecimiento semántico realizado por el modelo LLM. Campos principales: identificador, referencia a la oferta, habilidad normalizada, tipo de habilidad (explícita, implícita o normalizada), URI del concepto ESCO, nivel de confianza del LLM, justificación del razonamiento, y modelo utilizado.

skill_embeddings: Contiene las representaciones vectoriales de 768 dimensiones. Almacena identificador, texto de la habilidad, vector embedding con soporte pgvector, nombre del modelo, y versión. Incluye índice IVFFlat optimizado para búsquedas de similitud coseno con particionamiento en 100 listas.

analysis_results: Almacena resultados de clustering y análisis de tendencias. Campos: identificador, tipo de análisis (clustering, trends, profile), país, rango de fechas analizado, parámetros de configuración en formato JSONB, y resultados estructurados con clústeres, etiquetas y métricas.

esco_skills: Tabla de referencia con la taxonomía ESCO completa. Contiene URI del concepto, etiquetas preferidas en español e inglés, etiquetas alternativas, tipo de habilidad, descripción, y nivel de reutilización. Almacena 13,174 habilidades (13,000 de ESCO v1.1.0 más 174 agregadas manualmente).

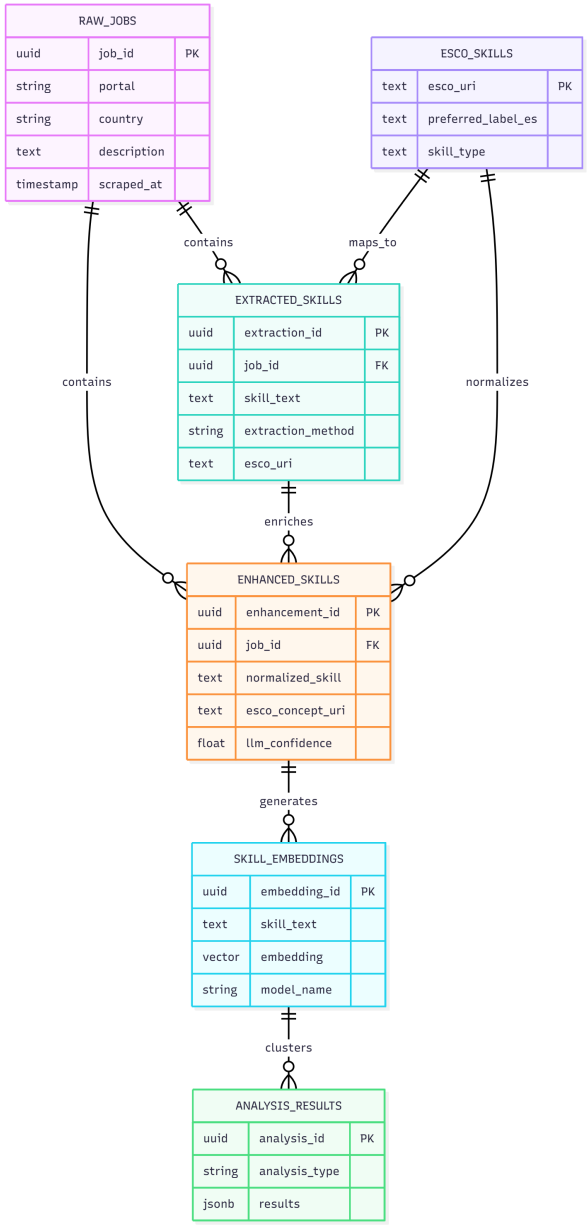

Figura 5.2: Diagrama Entidad-Relación de la Base de Datos

La Figura 5.2 muestra las relaciones entre tablas. Todas las tablas derivadas mantienen referencia mediante llave foránea hacia la tabla de ofertas laborales (raw_jobs), garantizando trazabilidad completa desde cualquier resultado de análisis hasta la oferta original. Las tablas de habilidades extraídas y enriquecidas mantienen referencia opcional hacia la taxonomía ESCO para efectos de normalización.

5.4 Herramientas y Tecnologías

Las Tablas 5.5 y 5.6 resumen las decisiones tecnológicas fundamentales y su justificación académica y técnica, organizadas por capa funcional del sistema.

Tabla 5.5: Stack Tecnológico: Infraestructura y Adquisición de Datos

Componente	**Tecnología**	**Justificación**
Base de datos	PostgreSQL 15+ con pgvector	Soporte JSONB para metadatos flexibles, extensión pgvector para vectores 768D, robustez ACID, particionamiento para escalabilidad.
Taxonomía	ESCO v1.1.0 (+ 174 manuales)	Cobertura 13,000+ skills con etiquetas español/inglés, estructura ontológica con URIs, respaldo institucional CE, licencia CC BY 4.0.
Framework scraping	Scrapy 2.11 + Selenium 4.15	Arquitectura asíncrona (100+ req/min), manejo robusto de reintentos, middlewares extensibles, Selenium para JavaScript dinámico.
Modelo NLP español	spaCy 3.7 + es_core_news_lg	Mejor modelo español disponible (97M parámetros), soporte EntityRuler para ESCO, optimizado CPU ($<$100ms/doc).
Lenguaje	Python 3.11+	Ecosistema científico maduro (NumPy, pandas, scikit-learn), bibliotecas NLP referencia (spaCy, transformers), integración PostgreSQL.
Control versiones	Git + GitHub	Estándar industria, integración CI/CD (GitHub Actions), control issues/milestones, documentación Markdown, respaldo cloud.

Tabla 5.6: Stack Tecnológico: Procesamiento y Análisis

Componente	**Tecnología**	**Justificación**
LLM enriquecimiento	LLaMA 3 8B Instruct (Q4)	F1=0.64 en español técnico, despliegue local sin APIs (privacidad), cuantización Q4 (8-10 GB VRAM), tokenización eficiente 1.3 tokens/palabra.
Embeddings	intfloat/multilingual-e5-base	Estado del arte multilingüe (768D), contrastive learning en 100 idiomas, normalización L2 integrada, 278M parámetros ejecutables CPU.
Índice similitud	FAISS IndexFlatIP	Velocidad 30,147 q/s (25x superior pgvector), búsqueda exacta (100 % recall), latencia 0.033ms, Facebook AI Research.
Reducción dimensional	UMAP [41]	Preserva estructura local y global (superior t-SNE), escalabilidad millones puntos, reproducibilidad con semilla fija, parámetros interpretables.
Clustering	HDBSCAN [42]	No requiere especificar k, identifica ruido automático, maneja densidades variables (nicho vs. mainstream), jerarquía multinivel.
Orquestación	Typer CLI + APScheduler	Interface tipo Git con validación automática, scheduler 24/7, logging estructurado, integración cron/systemd.

Todas las tecnologías seleccionadas cumplen con criterios de: (1) Licencias permisivas (MIT, Apache 2.0, PostgreSQL, CC BY) permitiendo uso académico y potencial comercial futuro; (2) Madurez y estabilidad con versiones $\geq$ 2.0 y comunidades activas; (3) Documentación académica completa con publicaciones científicas revisadas por pares para componentes críticos; (4) Reproducibilidad mediante control de versiones de dependencias y semillas fijas para componentes estocásticos; (5) Escalabilidad demostrada para el objetivo de 600,000 ofertas laborales mediante arquitectura de pipeline por lotes y particionamiento de base de datos.

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.

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.

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.

REFERENCIAS

[1] O. Azuara et al., "COVID-19 y el mercado laboral en América Latina: diagnóstico y políticas," Banco Interamericano de Desarrollo, 2022.

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

[3] J. F. Rubio Arrubla, "Demanda de habilidades tecnológicas: evidencia desde el mercado laboral colombiano," Universidad de los Andes, Centro de Estudios sobre Desarrollo Económico (CEDE), Documento CEDE 2025-18, jun. de 2025.

[4] M. Aguilera y S. Méndez, "Análisis del mercado laboral TI en Argentina mediante web scraping," Proyecto de Grado, Universidad del Sinú - Seccional Cartagena, 2018. dirección: http://repositorio.unisinucartagena.edu.co:8080/jspui/bitstream/123456789/94/1/1.%20Proyecto%20de%20Grado%20II%20-%20WEB%20SCRAPING_FINAL.pdf

[5] C. Martínez Sánchez, "Demanda de habilidades digitales en México: un análisis empírico," Tesis de mtría., UNAM, 2024.

[6] J. Cárdenas Rubio et al., "Análisis del mercado laboral colombiano mediante técnicas de minería de texto," *Revista Colombiana de Computación*, 2015.

[7] R. Campos-Vázquez y C. Martínez Sánchez, "Skill Mismatch in the Mexican Labor Market," en *Proceedings of the Labor Economics Conference*, 2024.

[8] M. Lukauskas, V. Šarkauskaitė, V. Pilinkienė, A. Stundžienė, A. Grybauskas y J. Bruneckienė, "Enhancing skills demand understanding through job ad segmentation using NLP and clustering techniques," *Applied Sciences*, vol. 13, n.º 10, pág. 6119, mayo de 2023. DOI: 10.3390/app13106119

[9] A. Herandi, Y. Li, Z. Liu, X. Hu y X. Cai, *Skill-LLM: Repurposing general-purpose LLMs for skill extraction*, oct. de 2024. DOI: 10.48550/arXiv.2410.12052 arXiv: 2410.12052.

[10] K. C. Nguyen, M. Zhang, S. Montariol y A. Bosselut, "Rethinking Skill Extraction in the Job Market Domain using Large Language Models," en *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*, E. Hruschka, T. Lake, N. Otani y T. Mitchell, eds., Association for Computational Linguistics, 2024, págs. 27-42. DOI: 10.18653/v1/2024.nlp4hr-1.3

[11] D. C. Kavargyris, K. Georgiou, E. Papaioannou, K. Petrakis, N. Mittas y L. Angelis, "ESCOX: A tool for skill and occupation extraction using LLMs from unstructured text," *Software Impacts*, jun. de 2025. DOI: 10.1016/j.simpa.2025.100772

[12] C. Orozco Puello y H. Gómez Estrada, "Web Scraping: técnicas y aplicaciones para análisis de datos," *Revista Colombiana de Tecnologías de Avanzada*, 2019.

[13] M. Zhang, K. N. Jensen, S. D. Sonniks y B. Plank, "SKILLSPAN: Hard and Soft Skill Extraction from English Job Postings," en *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2022, págs. 4962-4984. DOI: 10.18653/v1/2022.naacl-main.366

[14] H. Kavas, M. Serra-Vidal y L. Wanner, "Enhancing job posting classification with multilingual embeddings and large language models," en *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, Pisa, Italia, 2024, págs. 440-450. DOI: 10.18653/v1/2024.clicit-1.53

[15] S. Sarawagi, "Information Extraction," *Foundations and Trends in Databases*, vol. 1, n.º 3, págs. 261-377, 2008. DOI: 10.1561/1900000003

[16] D. Nadeau y S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, n.º 1, págs. 3-26, 2007. DOI: 10.1075/li.30.1.03nad

[17] J. Devlin, M.-W. Chang, K. Lee y K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," en *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, Association for Computational Linguistics, 2019, págs. 4171-4186. DOI: 10.18653/v1/N19-1423

[18] Y. Zhang, V. Zhong, D. Chen, G. Angeli y C. D. Manning, "Position-aware Attention and Supervised Data Improve Slot Filling," en *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, Association for Computational Linguistics, 2017, págs. 35-45. DOI: 10.18653/v1/D17-1004

[19] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang y J. Pérez, "Spanish Pre-Trained BERT Model and Evaluation Data," en *Proceedings of PML4DC at ICLR 2020*, 2020. dirección: https://github.com/dccuchile/beto

[20] J. De Corte, S. Vandevelde, M. Van de Kerkhof y L. Vanhee, "Neural Skill Extraction from Job Descriptions using Pre-trained Language Models," en *Proceedings of the 2021 IEEE International Conference on Big Data*, IEEE, 2021, págs. 2784-2793. DOI: 10.1109/BigData52589.2021.9671376

[21] J. E. F. Friedl, *Mastering Regular Expressions*, 3rd. O'Reilly Media, 2006, ISBN: 978-0596528126.

[22] L. Chiticariu, Y. Li y F. R. Reiss, "Rule-Based Information Extraction is Dead! Long Live Rule-Based Information Extraction Systems!" En *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, Association for Computational Linguistics, 2013, págs. 827-832. dirección: https://aclanthology.org/D13-1079/

[23] T. B. Brown et al., "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems (NeurIPS 2020)*, vol. 33, págs. 1877-1901, 2020. dirección: https://arxiv.org/abs/2005.14165

[24] H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," *arXiv preprint arXiv:2302.13971*, feb. de 2023. dirección: https://arxiv.org/abs/2302.13971

[25] C. Zhang, Z. Li, H. Wang, Y. Yang, Y. Liu y W. Wang, *Evaluating Large Language Models for Skill Extraction from Job Descriptions*, 2023. DOI: 10.48550/arXiv.2311.09213 arXiv: 2311.09213.

[26] J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," *Advances in Neural Information Processing Systems (NeurIPS 2022)*, vol. 35, págs. 24 824-24 837, 2022. dirección: https://arxiv.org/abs/2201.11903

[27] D. Vilares, M. A. Alonso y C. Gómez-Rodríguez, "EN-ES-CS: An English-Spanish Code-Switching Twitter Corpus for Multilingual Sentiment Analysis," en *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia: European Language Resources Association, 2016, págs. 4149-4153.

[28] Y. Elazar, J. Baan, L. Schut et al., *The Truth is in There: Improving Reasoning in Language Models with Layer-Selective Rank Reduction*, 2023. DOI: 10.48550/arXiv.2312.13558 arXiv: 2312.13558.

[29] Z. Ji et al., "Survey of Hallucination in Natural Language Generation," *ACM Computing Surveys*, vol. 55, n.º 12, págs. 1-38, 2023. DOI: 10.1145/3571730

[30] M. Bañón et al., "ParaCrawl: Web-Scale Acquisition of Parallel Corpora," en *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, Association for Computational Linguistics, 2020, págs. 4555-4567. DOI: 10.18653/v1/2020.acl-main.417

[31] J. Li, A. Sun, J. Han y C. Li, "A Survey on Deep Learning for Named Entity Recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, n.º 1, págs. 50-70, 2022. DOI: 10.1109/TKDE.2020.2981314

[32] A. Vaswani et al., "Attention is All You Need," en *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, I. Guyon et al., eds., Curran Associates, Inc., 2017, págs. 5998-6008. dirección: https://papers.nips.cc/paper/7181-attention-is-all-you-need

[33] J. Wei et al., "Emergent Abilities of Large Language Models," *Transactions on Machine Learning Research (TMLR)*, oct. de 2022. dirección: https://arxiv.org/abs/2206.07682

[34] OpenAI, "GPT-4 Technical Report," OpenAI, inf. téc., mar. de 2023, arXiv:2303.08774. dirección: https://arxiv.org/abs/2303.08774

[35] A. Q. Jiang et al., *Mistral 7B*, oct. de 2023. DOI: 10.48550/arXiv.2310.06825 arXiv: 2310.06825.

[36] Meta AI, *Llama 3 Model Card*, Released April 2024, abr. de 2024. dirección: https://github.com/meta-llama/llama3

[37] T. Mikolov, I. Sutskever, K. Chen, G. Corrado y J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," en *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, 2013, págs. 3111-3119. dirección: https://arxiv.org/abs/1310.4546

[38] N. Reimers e I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," en *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, Association for Computational Linguistics, 2019, págs. 3982-3992. DOI: 10.18653/v1/D19-1410

[39] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder y F. Wei, "Multilingual E5 Text Embeddings: A Technical Report," *arXiv preprint arXiv:2402.05672*, feb. de 2024. dirección: https://arxiv.org/abs/2402.05672

[40] J. Johnson, M. Douze y H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, n.º 3, págs. 535-547, 2021, FAISS library reference. DOI: 10.1109/TBDATA.2019.2921572

[41] L. McInnes, J. Healy y J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," *arXiv preprint arXiv:1802.03426*, feb. de 2018, Published in Journal of Open Source Software, 3(29), 861. DOI: 10.48550/arXiv.1802.03426 dirección: https://arxiv.org/abs/1802.03426

[42] R. J. G. B. Campello, D. Moulavi y J. Sander, "Density-Based Clustering Based on Hierarchical Density Estimates," en *Advances in Knowledge Discovery and Data Mining (PAKDD 2013)*, ép. Lecture Notes in Computer Science, Original HDBSCAN algorithm paper, vol. 7819, Springer, 2013, págs. 160-172. DOI: 10.1007/978-3-642-37456-2_14

APÉNDICES

En esta sección del documento se presenta la lista de todos los apéndices relacionados con el proyecto. Los apéndices deben ser descargables desde el sitio web y deben coincidir con los especificados en la propuesta.