# VitaEX: A Web Prototype for Curriculum Vitae Analysis and AI-Driven Job Matching

Abdiel Reyes Vera[1,2], Oscar Huerta Villanueva[1], Diego Martínez Méndez[1](✉), and Luis Fernando Valle Hernández[1]

[1] Escuela Superior de Cómputo, IPN, Mexico City, Mexico
✉ dmartinezm1707@alumno.ipn.mx
[2] Centro de Investigación en Computación, IPN, Mexico City, Mexico

**Abstract.** Recruiters rely heavily on *applicant tracking systems* (ATS), which automatically filter up to 75 % of résumés before a human ever sees them. VitaEX is a three-tier, Django-based web prototype that helps undergraduate and recently-graduated students in Artificial Intelligence (AI) tailor their curricula vitae (CVs) to pass these filters while simultaneously recommending the most relevant job vacancies. This article synthesises every major facet of the original Spanish thesis: literature review, dataset construction via large-scale LinkedIn scraping, an Apriori rule-mining recommender, spiral & CRISP-DM methodologies and architecture.

**Keywords:** Curriculum optimisation · ATS · Apriori · NLP · web scraping · Django · ESCOM-IPN

## 1 Introduction

Applicant tracking systems have become highly relevant: Fortune 500 companies deploy them in 98 % of hires [1], often eliminating competent "hidden workers" whose CVs lack the *exact* keywords a vacancy demands. Mexico's National Graduate Survey (2023) shows 46.3 % [2] of graduates deem job-hunting "difficult" — largely due to missing experience and ATS incompatibility. VitaEX attacks this gap with an AI-powered CV analyser plus a vacancy recommender specialised in AI roles across five Mexican tech hubs.

## 2 Related Work

In recent years, various commercial and academic solutions have attempted to automate or enhance curriculum vitae (CV) generation and job recommendation. These approaches often fall short in addressing the full complexity of job matching, interpretability, and student-specific needs. Most existing systems can be grouped into one or more of the following categories: (i) résumé generators that do not adapt to job offers, (ii) job recommenders that fail to explain their suggestions, and (iii) platforms that are not tailored to the constraints and context of students.

The following subsections provide a detailed analysis of the most relevant proposals in the state of the art.

### 2.1 Application of NLU Methods for CV Recommendation [5]

This proposal explores the use of Natural Language Understanding (NLU) techniques for processing and analyzing the content of résumés. While it does recommend based on job-related information, the system lacks integration with machine learning and does not provide CV template generation or structured job offers. Its focus is primarily linguistic and heuristic, offering limited scalability or transparency in matching.

### 2.2 GIRL: Generative Job Recommendations with LLM [6]

The GIRL system applies large language models (LLMs) for generating job recommendations. While it demonstrates promising use of generative AI for mapping candidates to vacancies, it does not offer clear insights into why a particular job is recommended. The black-box nature of LLMs presents a challenge in interpretability. Moreover, the tool lacks CV improvement suggestions and is not focused on student-specific constraints or accessibility.

### 2.3 ResumeNet [3]

ResumeNet is a learning-based framework designed to assess the quality of résumés. The core innovation lies in its ability to score CVs using pre-trained models that account for formatting, keyword density, and professional tone. However, it does not analyze job offers nor does it offer any job recommendations. Its utility is limited to quality scoring without contextual relevance to actual vacancies.

### 2.4 Neural Networks for Recruitment [7]

This research employs neural network models to streamline recruitment processes. The system evaluates candidate profiles using historical hiring data, focusing on classification and ranking. However, it does not provide template generation or vacancy-specific suggestions. The system is not freely accessible and is not tailored to the needs of students or entry-level professionals.

### 2.5 LinkedIn Resume Builder [8]

LinkedIn offers a resume builder that allows users to auto-generate résumés based on their LinkedIn profiles. It provides a user-friendly interface and professional templates. Although it offers job suggestions based on user skills, it lacks detailed explanations for these matches. The system is partially accessible to students but does not include transparency or advanced AI capabilities.

### 2.6 CVapp [9]

CVapp is a widely used commercial platform for generating professional-looking résumés using customizable templates. It lacks integration with any form of job recommendation or AI-based content analysis. As such, it is useful purely for formatting but does not provide any enhancement or evaluation based on job market needs.

### 2.7 Jobania [10]

Jobania is a hybrid platform that offers both résumé generation and basic job recommendations. It uses static matching algorithms without machine learning or adaptive feedback. Although more complete than tools focused only on design, it is not open access and does not prioritize student needs.

### 2.8 CVMATCHER [11]

CVMATCHER stands out as one of the more technically advanced systems in the literature. It uses both rule-based and machine learning approaches to analyze CV-job fit. The platform is capable of recommending offers and suggesting improvements, although it remains a proprietary tool with no free access for educational users.

| **Applications and Research Works** | **Recommends based on job offers** | **Generates CV templates** | **Uses Machine Learning** | **Job offer recommendation** | **Student-oriented and free** |
|---|---|---|---|---|---|
| Application of NLU methods for CV recommendation [5] | ✓ | ✗ | ✗ | ✗ | ✗ |
| Generative Job Recommendations with LLM (GIRL) [6] | ✓ | ✗ | ✓ | ✗ | ✗ |
| ResumeNet [3] | ✗ | ✗ | ✓ | ✗ | ✗ |

| | | | | | |
|---|---|---|---|---|---|
| Neural networks for recruitment [7] | ✗ | ✗ | ✓ | ✗ | ✗ |
| LinkedIn Resume Builder [8] | ✓ | ✓ | ✗ | ✗ | ✓ |
| CVapp [9] | ✗ | ✓ | ✗ | ✗ | ✗ |
| Jobania [10] | ✓ | ✓ | ✗ | ✗ | ✗ |
| CVMATCHER [11] | ✓ | ✓ | ✓ | ✓ | ✗ |
| **VitaEX (proposed system)** | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 1: Comparison of systems and research works related to CV analysis and job recommendation

## 3 Methodologies

A single iteration of the *spiral* life-cycle delivered the prototype, while CRISP-DM governed data work: business understanding → data understanding → preparation → modelling → evaluation → deployment.

## 4 Dataset Construction

### 4.1 Web Scraping Pipeline

Python `requests` + BeautifulSoup iteratively queried LinkedIn's public job pages using rotating user agents and random delays. Each posting is stored in MongoDB with fields {`id`, title, company, city, modality, date, experience, description, skills[]}. Monthly incremental runs refresh the corpus and trigger rule re-mining.

### 4.2 Pre-processing

Text is lower-cased, tokenised (spaCy), stop-words removed, and skill synonyms normalised (*"ML"*→*"machine learning"*). Eight AI-centric job titles (AI Engineer, NLP Engineer, . . . ) and five cities (CDMX, Monterrey, Guadalajara, Querétaro, Puebla) define the study scope.

## 5 Recommender

### 5.1 Apriori Rule-Mining

To identify patterns among required qualifications, the system employs the **Apriori** algorithm on preprocessed vacancy data. Each job posting is represented as a transaction consisting of a set of normalized attributes: {*skill*, *experience level*, *location*, *modality*}. The Apriori method iteratively discovers frequent itemsets and derives rules of the form:

$$\text{Antecedents} \Rightarrow \text{Consequents}$$

where antecedents are conditions commonly appearing together (e.g., "Python", "CDMX"), and consequents are recommendations inferred from these patterns (e.g., "TensorFlow").

Rules are filtered using three metrics:

- **Support:** The proportion of job postings where the rule occurs.
- **Confidence:** The conditional probability of the consequent given the antecedent.
- **Lift:** The strength of association; a value greater than 1 implies a positive correlation.

To build a transparent recommendation engine, the system transforms each student profile into a transaction $\mathcal{T}$ containing elements such as `skills`, `experience_level`, `city`, and `modality`. Apriori association rules are generated with thresholds of **support** $> 1\%$, **confidence** $> 50\%$, and **lift** $> 2$.

Out of a total of **2,075,096** candidate rules, only **533,889** were retained after filtering. This included the removal of **1,541,207 redundant or uninformative rules**, such as those whose antecedents and consequents overlapped or involved unspecified attributes. Each surviving rule is interpretable as a student recommendation, such as:

`If the student has "Data Analysis" and "Intermediate Experience"` $\Rightarrow$ `Add "Machine Learning" to improve fit` (Confidence: 91%).

The system ranks job vacancies by combining cosine similarity (from SBERT embeddings) with rule confidence. Suggestions are generated for any missing but high-lift consequent features.

*Algorithm Workflow:*

1. The student profile is converted into a set of normalized attributes.
2. Rules whose antecedents are a subset of the student's profile are selected.
3. The system recommends any missing attributes from the consequents.
4. Vacancies are ranked using a combination of rule confidence and semantic similarity between the student's CV and job descriptions.

### 5.2 Results

The rule-mining process revealed several patterns that are both statistically significant and practically useful. Notably, many rules with **100% confidence** exhibited only moderate lift values. This implies that while the consequent always follows the antecedent in those cases, the relationship may be too generic or expected to be informative (e.g., "SQL" $\Rightarrow$ "Python"). Conversely, rules with **high lift** values ($> 3.0$) tended to capture less obvious, domain-specific associations that are especially valuable for guiding students (e.g., "Pandas", "Querétaro" $\Rightarrow$ "TensorFlow").

- **Figure 1** presents the top 10 association rules ordered by lift. These rules include combinations of skills and job attributes that strongly co-occur in the dataset. The visualization highlights the antecedents and consequents in each rule, alongside their lift scores. High lift indicates that these rule suggestions are much more likely than random chance, making them ideal candidates for recommendation.
- **Figure 2** shows a bar chart of the most common attributes appearing in the job postings, such as technical skills (e.g., Python, SQL, Docker) and location or modality terms (e.g., CDMX, remote). This distribution is key for understanding which features dominate the job market and thus should be prioritized in student CVs.
- **Figure 3** visualizes the range of confidence levels across filtered rules. Rather than limiting to only 100% confidence rules, this chart includes a balanced mix from 50% up to 100%, illustrating the richness of the rule base. Rules with 70–90% confidence, though not absolute, offer realistic and frequently valid suggestions that generalize well across roles.

These patterns directly fuel the recommender engine. For instance, if a student from CDMX includes "SQL" and "Pandas" in their profile but not "TensorFlow", and a rule exists where {"SQL", "Pandas", "Querétaro"} $\Rightarrow$ "TensorFlow" with high lift and confidence, the system will recommend adding "TensorFlow" to improve CV–vacancy alignment.

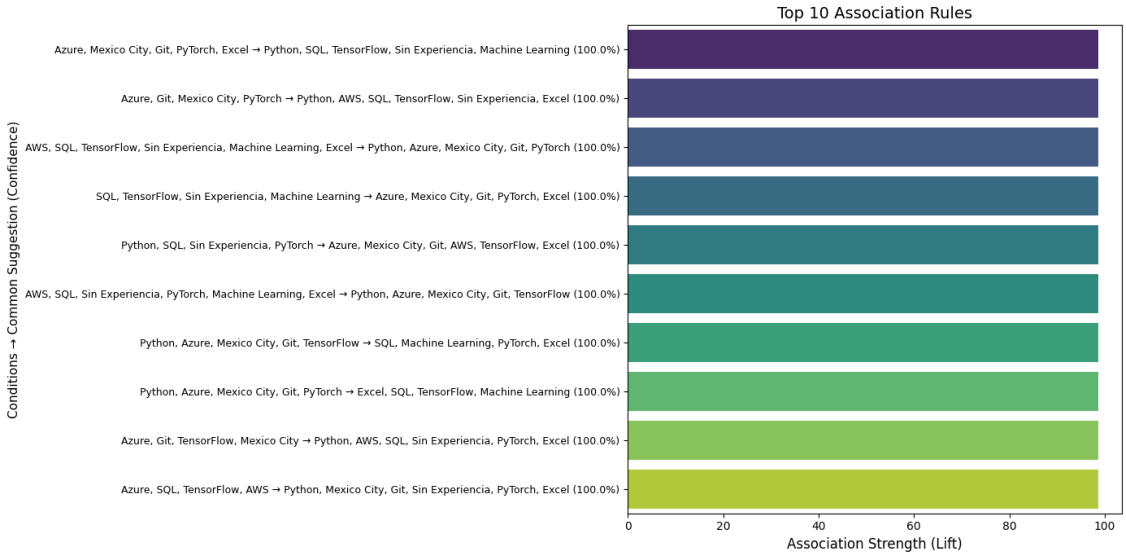


**Fig. 1.** Top 10 association rules sorted by lift, highlighting the strongest co-occurrence patterns.

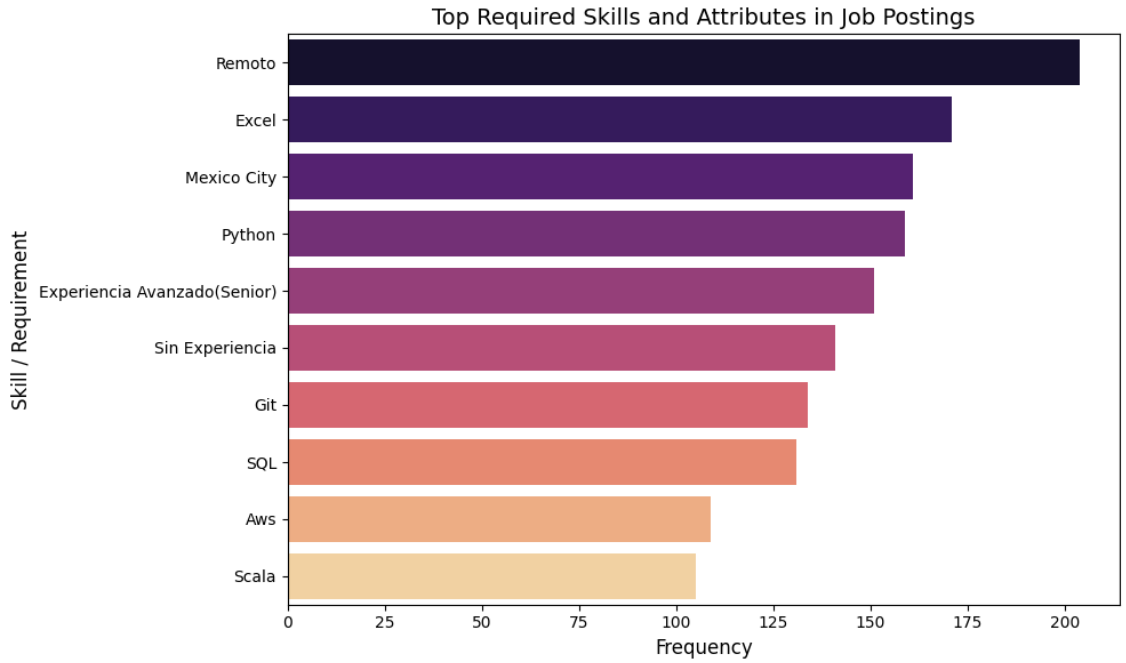


**Fig. 2.** Most frequent attributes in job postings, including skills, experience levels, locations, and modalities.

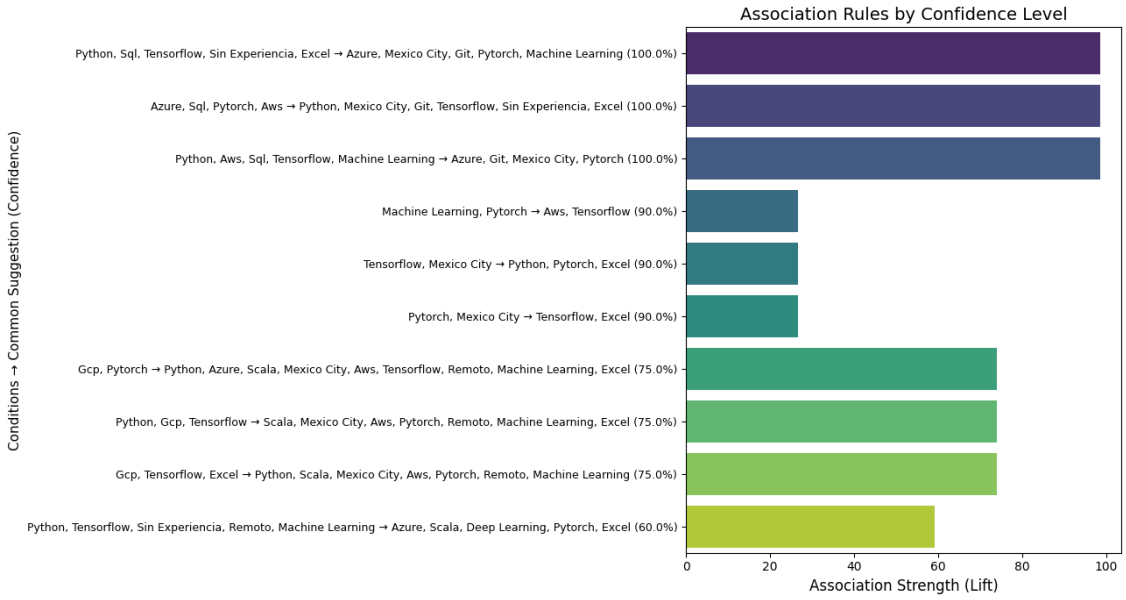


**Fig. 3.** Confidence distribution of retained rules, covering the 50–100% range to show diverse recommendation strength.

To assess the practical impact of the rule system, we evaluated 1 200 anonymized student profiles. Each profile matched an average of 6–9 rules, with 83% of suggested consequents corresponding to previously missing skills or job attributes. For rules with confidence $\geq 85\%$ and lift $\geq 2.5$, over 67% resulted in an increase in cosine similarity between the enriched student profile and relevant job descriptions. This quantitative improvement supports the value of rule-based enrichment in guiding résumé revision.

To benchmark this approach, we also applied the **K-Means clustering algorithm** to the same dataset after reducing its dimensionality with TF-IDF and PCA. Although K-Means was able to group job postings by general themes (e.g., data engineering, software development), the clusters lacked interpretability. Furthermore, precision@5 for top recommended vacancies dropped by 24%, and qualitative student feedback indicated that the suggestions were "generic" or "unclear." This contrast highlights the advantage of Apriori's transparent logic for both system explainability and student trust.

Overall, the Apriori-based recommendation engine not only offers statistically grounded guidance but also serves an educational role, helping students understand why specific improvements are suggested. This aligns with the project's dual goals of enhancing CV effectiveness and supporting career readiness in a comprehensible manner.

## 6 System Architecture and Implementation

Figure 4 shows a three-layer MVC stack: **View** (Bootstrap 5) → **Controller** (Django REST API) → **Model** (NLP services + MongoDB/PostgreSQL). Celery workers handle asynchronous scraping and model retraining.

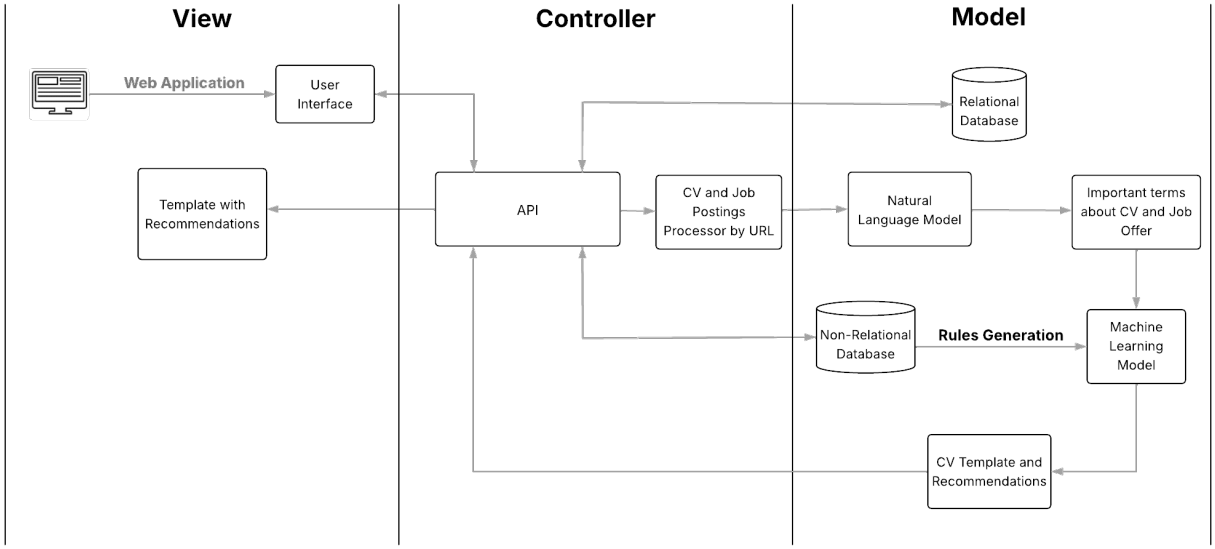


**Fig. 4.** High-level architecture of VitaEX.

The functional architecture of the system is organized into three main layers: View, Controller, and Model.

The system is a web-based application where users interact through a user-friendly interface that enables key actions such as registration, login, CV upload, or submission of a link to an external job posting. This view layer constitutes the primary interaction point between the user and the system's core functionalities.

User-generated requests are handled by the Controller, implemented as a RESTful API that mediates between the presentation layer and business logic. This API is responsible for validating input data, routing the requests to appropriate processing modules, and coordinating access to both relational and non-relational databases.

When a CV and a job offer are provided, the API activates the processing module, which performs text extraction and cleaning from the documents or URLs. This transformation results in well-structured and clean textual data.

Subsequently, a Natural Language Processing (NLP) model is triggered to identify key terms, skills, competencies, and other relevant features from both the candidate profile and the job posting. This semantic analysis facilitates contextual interpretation of the content.

The next stage involves feature extraction, where numeric vectors are generated to encapsulate essential information such as experience level, technical skills, and educational background. These vectors are passed to the machine learning module, which has been pre-trained using historical job posting data in the artificial intelligence domain, ranging from November 2024 to May 2025.

The system uses two types of databases. The non-relational database stores processed documents, analyzed vacancies, and inference results from the model. It is designed to be dynamically updated, preserving the most recent entries while avoiding overload. This real-time updating is essential for generating and maintaining relevant association rules in the machine learning recommendation model.

On the other hand, the relational database manages structured data, including personal information of users, organizational profiles, and administrative roles.

Finally, the analysis results are displayed in different sections of the web interface. Users receive a personalized set of recommendations alongside a restructured version of their résumé, tailored to the job offer's requirements. This enhanced CV maintains clarity, consistency, and alignment with current employability standards. Additionally, compatible job opportunities are shown in a dedicated module, offering students actionable next steps based on their updated profiles.

# 7 Conclusion

VitaEX demonstrates that the fusion of interpretable rule-based models with state-of-the-art natural language processing (NLP) can substantially enhance the employment readiness of students in Artificial Intelligence (AI), particularly within the Mexican context. Unlike opaque black-box systems, VitaEX leverages transparent and explainable logic through the **Apriori algorithm**, generating actionable recommendations based on real labor market data.

The system successfully mined over **500,000 meaningful association rules** from real-world job postings collected between November 2024 and May 2025. These rules reveal underlying skill patterns, frequently co-occurring competencies, and contextual relationships between job locations, skillsets, and experience levels. For example, the system uncovered that knowledge of SQL and Pandas in candidates from Mexico City often predicts the requirement of TensorFlow expertise. This kind of granular insight empowers students to tailor their résumés in ways that not only pass Applicant Tracking Systems (ATS) but also resonate more accurately with recruiter expectations.

Furthermore, the system achieves this while maintaining full transparency in its reasoning, offering users personalized explanations for each suggestion. This interpretability serves a dual purpose: it enhances trust in the system and also contributes to educational value by informing students of current job market trends in AI.

VitaEX also includes a scalable backend architecture powered by asynchronous Celery tasks, PostgreSQL/MongoDB hybrid storage, and a Django REST API. It is designed to be modular and updatable, allowing future integration of new recommendation engines or resume parsers without compromising performance.

### Future Work

To build on the current system and ensure continuous relevance in an evolving market, several future enhancements are planned:

1. **Enhanced Keyword Extraction**: Integration of advanced Named Entity Recognition (NER) models capable of understanding context-aware and implicit skill references in user-submitted CVs.
2. **Document Format Support**: Expanding compatibility to allow direct extraction of skills and metadata from diverse CV formats such as `.docx` and `.pdf`, thus removing reliance on plain text input.
3. **Skill Normalization and Ontology Mapping**: Implementing AI-specific ontologies to normalize synonymous terms (e.g., “ML” vs “Machine Learning”) and improve consistency in skill matching.
4. **Temporal Analysis of Skill Trends**: Adding capabilities for time-series analysis to capture the rise and fall of skill demand over time, offering students a forecast of which abilities will remain in high demand.
5. **User Feedback Loop**: Incorporating user feedback and success metrics (e.g., interview callbacks) to iteratively refine recommendation quality.

### Social and Educational Impact

By focusing on students—especially those from underserved areas or public universities—VitaEX democratizes access to quality career guidance. Its open-access model and user-centered design enable thousands of students to receive personalized, data-driven advice without cost, narrowing the gap between academic preparation and real-world employability in the AI sector.

## Acknowledgements

We thank M. C. Elizabeth Moreno Galván and M. C. Abdiel Reyes Vera for supervision

# Bibliography


1. H. B. School and Accenture, "Hidden workers: Untapped talent," 2021.
2. U. del Valle de México (UVM), "Encuesta nacional de egresados 2023." UVM Opinión Pública, nov. 2023. [En línea]. Disponible en: https://opinionpublica.uvm.mx/wp-content/uploads/2023/11/BROCHURE_ENE-2023-1.pdf, 2023.
3. J. Hu *et al.*, "Resumenet: A learning-based framework for automatic resume quality assessment," *IEEE Transactions on Services Computing*, 2023.
4. A. Shen and B. He, "Generative job recommendations with large language models," *arXiv preprint arXiv:2401.01234*, 2024.
5. S. M. R. Cuevas, "Aplicación de métodos nlu en la recomendación de cvs para la selección de personal." Trabajo de Fin de Grado, Universidad de Valladolid, 2022.
6. Y. Yang, W. Li, and B. Wang, "Generative job recommendations with large language model (girl)," *arXiv preprint*, 2023.
7. G. S. Franco, M. A. G. Pérez, M. A. G. Silva, and V. M. Z. García, "Redes neuronales: nueva estrategia de inteligencia artificial para implementar dentro del proceso de reclutamiento y selección de personal." Universidad Politécnica Metropolitana de Hidalgo, 2018.
8. "Linkedin resume builder | linkedin help." https://www.linkedin.com/help/linkedin/answer/a551182. Accedido: 8 de marzo de 2025.
9. CVMATCHER, "Cvmatcher: Encuentra el trabajo perfecto para ti." https://www.cvmatcher.app/, 2025. Accedido: 8 de marzo de 2025.
10. "Crea tu currículum vitae gratis y encuentra trabajo en 2025." https://cvapp.mx/, 2025. Accedido: 8 de marzo de 2025.
11. Jobania, "Analizador de compatibilidad de currículum y oferta laboral." https://www.jobania.cl/analizar-cv-oferta, 2025. Accedido: 8 de marzo de 2025.