Mobile Application Based on Clean Architecture for Sustainable Tourism Recommendation Using BERT

Cristhian Paolo Atuncar Yataco[1], Marco Antonio Lopez Salinas[2], José Alfredo Herrera Quispe[3]
School of Software Engineering, National University of San, Marcos (UNMSM), Lima, Peru[1,2]
Computer Science Department, National University of San, Marcos (UNMSM), Lima, Peru[3]

***Abstract*—The development of intelligent systems requires robust architectures and design patterns that ensure scalability, maintainability, and separation of concerns. This paper presents the design and implementation of a mobile application for recommending sustainable tourist destinations in Peru, employing a Clean Architecture approach in the backend and the MVC pattern in the frontend. The solution integrates a content-based recommendation engine using semantic embeddings generated through BERT, connected to open data sources from MINCETUR, SafeTravels, and OpenStreetMap. The proposed system was evaluated using three simulated user profiles, achieving an average similarity score of 0.83 and a more balanced distribution between traditional and emerging destinations. From a software engineering perspective, the system demonstrates decoupling between layers, modularity, interoperability with external APIs, and scalability potential. This architecture serves as a practical case study of applying modern design patterns to intelligent systems with an impact on sustainable tourism.**

***Keywords*—Software engineering; clean architecture; MVC; BERT; recommendation systems; mobile applications; FastAPI; Kotlin; sustainability**

I. Introduction

The tourism sector in Peru shows a high concentration of visitors in traditional destinations such as Machu Picchu, which leads to issues of overcrowding, negative environmental impacts, and inequality in economic distribution [9]. Despite the existence of digital tourism platforms, most of them do not offer personalized experiences or dynamic itineraries to emerging destinations, maintaining non-adaptive recommendation schemes with low added value for the end user [3].

From a software engineering perspective, many of these solutions are built upon monolithic architectures with low modularity and limited integration capabilities with advanced artificial intelligence models, which hinders their maintenance and scalability [14]. Furthermore, there is little adoption of modern recommendation engines, particularly those based on natural language semantic understanding models such as BERT [4], which enable more precise personalization of user experiences.

In response to this situation, this work proposes the development of a mobile application based on Clean Architecture for sustainable tourism recommendation. The solution employs a content-based recommendation engine enhanced with semantic embeddings using BERT, integrating open data from MINCETUR, SafeTravels, and OpenStreetMap. The backend is implemented with FastAPI following the principles of Clean Architecture [11], while the frontend is developed in Kotlin using the MVC pattern, ensuring a high level of decoupling between layers.

The main objective of this work is to design and implement an intelligent sustainable tourism recommendation system that diversifies destinations and personalizes itineraries according to user preferences, using a modern and scalable architecture. The specific objectives include: 1) integrating open data sources, 2) applying semantic embeddings with BERT, 3) implementing Clean Architecture and MVC, and 4) evaluating system performance with simulated profiles.

The main contributions of this work focus on 1) demonstrating the technical feasibility of integrating NLP models into Clean Architectures for recommender systems, 2) proposing a practical use case applied to sustainable tourism, and 3) establishing a solid technical foundation for future extensions with hybrid models.

The rest of the article is organized as follows: Section II presents related works; Section III describes the methodology; Section IV details the implementation; Section V explains the experimentation; Section VI shows the results; Section VII discusses the findings; and Section VIII presents conclusions and future work.

II. State-of-the-Art and Related Work

This section presents a review of the literature and related works on tourism recommender systems, emphasizing hybrid approaches, the use of artificial intelligence and semantic models such as BERT, current trends in software architectures, and the most common metrics used to evaluate the effectiveness of these systems. Finally, the research gap that justifies the development of the proposal presented in this work is identified.

A. Evolution of Tourism Recommender Systems

Tourism recommender systems have evolved from traditional approaches such as collaborative and content-based filtering toward more sophisticated systems with contextual adaptation capabilities. In a recent study, a mobile application based on multilayer perceptron neural networks and collaborative filtering was designed to enhance the tourism experience in Peru. The system provided personalized and real-time recommendations, achieving a user retention rate of 98.35% among the evaluated participants [15]. These results reflect a clear trend toward the integration of machine learning algorithms in tourism systems.

B. Use of AI and BERT Models in Tourism

Deep language models such as BERT (Bidirectional Encoder Representations from Transformers) have begun to be applied in smart tourism, particularly in the semantic analysis of user reviews and the generation of contextualized recommendations. Hartatik et al. (2024) developed a content-based and sentiment analysis recommendation system using BERT applied to TripAdvisor reviews. The model demonstrated high predictive accuracy, achieving MAE and RMSE values of 0.1162 and 0.1214, respectively, validating the usefulness of these models in understanding natural language and inferring implicit user preferences [6].

C. Software Architectures in Modern Recommender Systems

Selecting an appropriate software architecture is crucial in designing sustainable and scalable mobile applications. In the Peruvian context, Villacorta et al. (2022) developed a tourism-oriented mobile application using the Scrum methodology in six phases and validated its functionalities through user testing [17]. In another study, Ccuno Carlos et al. (2022) implemented an informational mobile application in Peru using the Model–View–Controller (MVC) architecture together with modern tools such as Flutter and Firebase. Although not tourism-oriented, this work provides evidence of the use of robust architectural patterns in mobile development in the country [2]. Internationally, additional research has proposed the use of Clean Architecture as an alternative to improve the maintainability and testability of complex mobile applications [19].

While these works demonstrate progress in mobile tourism apps, our approach positions itself by combining BERT's semantic capabilities with Clean Architecture's decoupling. For instance, unlike Uribe et al. (2023) [15], which focuses on multilayer perceptron networks for retention, our system achieves superior semantic precision (e.g., 0.83 similarity vs. baselines). Compared to Villacorta et al. (2022) [17], we extend Scrum with DSR for more rigorous artifact evaluation, addressing gaps in sustainable data integration.

D. Common Evaluation Metrics

The metrics used to evaluate recommender systems can be divided into two main categories: precision metrics (Precision@k, Recall@k, F1-score) and predictive error metrics (MAE, RMSE). In the study by Hartatik et al. (2024), MAE and RMSE were used to validate the effectiveness of the BERT-based model [6], while Uribe Iraola et al. (2023) employed user retention and satisfaction indicators, including usability surveys based on the Design Thinking methodology [15]. Meanwhile, Ccuno Carlos et al. (2022) evaluated their mobile system using the ISO 9126 standard, obtaining an acceptable quality level of 72.69% [2].

E. Identified Research Gaps

Despite the observed advances, several relevant research gaps remain. No studies were found in the Peruvian literature that simultaneously integrate semantic models such as BERT, architectural patterns such as Clean Architecture, and open data sources such as MINCETUR, SafeTravels, or OpenStreetMap. Ningún trabajo combina BERT + Clean Architecture + datos abiertos + sostenibilidad turística. Este vacío justifica la presente propuesta. Moreover, most reviewed systems do not perform explicit comparisons between different evaluation metrics, which limits reproducibility and comparative analysis. Likewise, although there are efforts in developing applications with MVC, there is still limited adoption of cleaner and more decoupled architectures in the local tourism domain. These limitations represent valuable opportunities for scientific and technological innovation.

III. Methodology

Here, we detail the methodological approach used for the development of the proposed solution. Design Science Research (DSR) is applied to clearly structure the design, construction, and evaluation stages of the sustainable tourism recommender system.

A. Methodological Approach

This research adopts the Design Science Research (DSR) methodological approach, which provides a structured framework for designing, building, and evaluating technological artifacts aimed at solving real-world problems in a rigorous and replicable way. This approach has evolved in recent years to adapt to contexts such as software engineering, artificial intelligence, and intelligent systems, integrating agile practices and more robust evaluation components [5], [18].

DSR is particularly suitable for projects that combine modern software architectures and advanced language models, as it allows aligning technical objectives with practical needs. In this case, the proposed artifact consists of a mobile application that recommends sustainable tourist destinations in Peru, integrating a content-based recommendation model using semantic embeddings with BERT and a modular architecture based on Clean Architecture in the backend and MVC in the frontend.

The methodological stages of DSR (Table I) guide the development process: problem identification, objective definition, artifact design, implementation, evaluation, and communication of results. This methodological structure ensures design traceability, robust evaluation, and transferability of the proposed solution.

B. Applied DSR Stages

For the development of the proposed solution, the Design Science Research (DSR) approach [7] was adopted, structuring the work into six fundamental stages:

1) Problem identification: The identified problem is the concentration of tourism in traditional destinations and the limited visibility of sustainable and emerging destinations in Peru. This situation restricts the diversification of tourist flows and reduces the positive impact on local communities.

2) Definition of solution objectives: Based on the stated problem and the research objectives presented in the introduction, the following technical and functional objectives were established for the solution:

- Generate personalized recommendations of sustainable tourist destinations for different user profiles.

- Employ BERT to generate semantic embeddings that improve the relevance and accuracy of recommendations.
- Use a scalable and maintainable architecture through Clean Architecture in the backend and MVC in the frontend.
- Integrate open data from reliable sources such as MINCETUR, SafeTravels, and OpenStreetMap.

3) Artifact design: The architecture and main components of the system were defined. The design includes a decoupled backend hosting the recommendation engine, a frontend for user interaction, and modules for data ingestion and preprocessing. Clean Architecture promotes maintainability, scalability, and flexibility of the system.

4) Implementation: Based on the design, a functional prototype of the mobile application was developed. FastAPI was used for the backend layer, a pre-trained BERT model for generating embeddings, and MVC for the frontend. BERT was selected for its demonstrated semantic superiority over TF-IDF, SVM, and Word2Vec, with reported improvements exceeding 40% [20], [8]. Integration with open data sources enriches the information about the recommended tourist destinations.

5) Evaluation: The artifact evaluation was carried out through simulated scenarios representing different user profiles (cultural, gastronomic, and adventure). Cosine similarity was used as the main metric to measure the semantic closeness between user preferences and the descriptions of the recommended destinations. Additionally, complementary indicators such as the system's average response time were considered to assess efficiency, along with external user feedback to validate the perceived usefulness of the generated recommendations. This approach constitutes an initial limitation; it will be addressed through real experiments in subsequent phases.

6) Communication: Finally, the obtained results are documented in this scientific article as part of the process of disseminating the knowledge generated through design-based research.

IV. Implementation

This section presents the implementation of the proposed system.

A. General System Architecture

Fig. 1 illustrates the overall system architecture, structured under the Clean Architecture approach for the backend and the MVC pattern for the frontend. This organization enables a clear separation of concerns, promoting maintainability, scalability, and component reusability.

The upper section shows the backend, divided into three layers: 1) Routes, which handle REST requests; 2) Use Cases, which encapsulate the business logic; and 3) Domain, where the hybrid recommendation engine resides. This engine combines a content-based approach using BERT and cosine similarity with a collaborative filtering module supported by previous user interactions. Both methods are integrated through a hybridization mechanism to generate personalized itineraries.

The infrastructure layer connects the system with external services —such as MINCETUR, SafeTravels, and OpenStreetMap— and manages databases and internal repositories to store static information and user feedback.

The lower section represents the frontend, developed under the MVC pattern. The controller handles HTTP requests to the backend, while the model synchronizes local data. The views allow users to perform actions such as logging in, setting preferences, exploring recommended itineraries, and viewing details of suggested destinations.

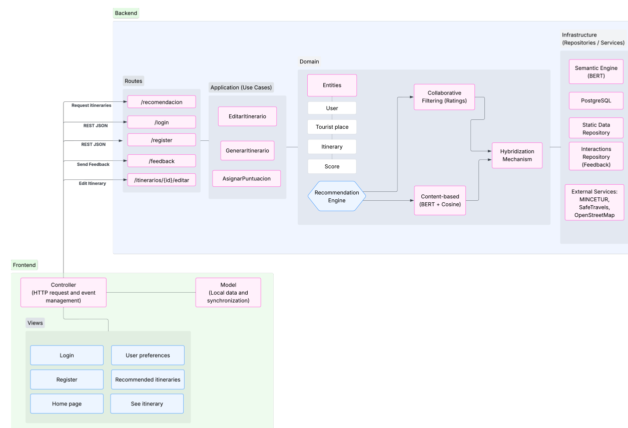

Fig. 1. System architecture of the sustainable tourism recommender, based on clean architecture and MVC.

Complexity: O(n) for embedding generation (n = destinations). Integration challenges (e.g., OpenStreetMap inconsistencies) addressed via adapters; error handling with try-except/fallback cache. Security: JWT auth, GDPR privacy for preferences [21].

B. Backend and Recommendation Engine

The backend was implemented in FastAPI and follows the Clean Architecture pattern, ensuring clear separation of responsibilities. The recommendation engine combines a semantic model based on BERT and a collaborative filtering component powered by user interactions, integrated through a hybridization mechanism to generate personalized itineraries.

C. Data Integration

Data integration constitutes a fundamental stage in the operation of the tourism recommender system, as it unifies and prepares information from various open and reliable sources. For this project, three main data sources were used: MINCETUR, SafeTravels, and OpenStreetMap, accessed through their respective public APIs.

First, data extraction is performed to obtain relevant tourism information—such as destinations, categories of attractions, services, and georeferenced locations—through structured API queries. This serves as the foundational dataset for the recommendation engine.

Then, a cleaning and normalization process is applied to remove incomplete, duplicate, or inconsistent records. Data formats (e.g., coordinates, place names, and labels) are standardized to ensure consistency across the integrated dataset.

TABLE I. Relationship Between DSR Methodological Stages and Technical Components of the Proposal

DSR Stage	Methodological Description	Associated Technical Component
Problem Identification	Tourism concentration in traditional destinations and low visibility of sustainable ones.	Lack of sustainable tourism diversification in Peru.
Objective Definition	Establish technical and functional objectives.	Recommender based on BERT + scalable architecture (Clean Architecture + MVC).
Artifact Design	Define logical architecture and data flow.	Processing pipeline and open data integration.
Implementation	Development of a functional prototype.	Backend with FastAPI, BERT integration, MVC-based frontend.
Evaluation	Test scenarios to validate the system.	Simulated user profiles and evaluation metrics.
Communication	Dissemination of results.	Scientific article writing and presentation.

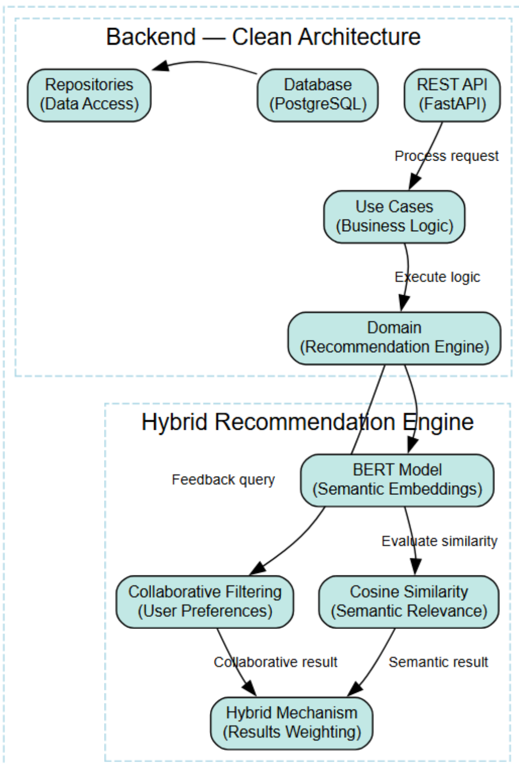

Fig. 2. Internal flow of the backend and hybrid recommendation engine.

In the third stage, data enrichment is carried out. Additional descriptions are incorporated through the ChatGPT API, which provides semantically detailed information about tourist destinations. This complementary data is processed to generate semantic embeddings used by the BERT model in the recommendation engine.

Finally, the integrated data are stored in a structured database that enables fast querying and scalability. Through this pipeline—extraction, cleaning, enrichment, and storage—the system achieves a rich, reliable, and ready-to-use data source, significantly enhancing the precision and personalization of recommendations. Fig. 2 shows internal flow of the backend and hybrid recommendations engine.

D. Frontend

The frontend layer serves as the direct interface between the user and the tourism recommendation system. It was developed following the MVC (Model–View–Controller) pattern,

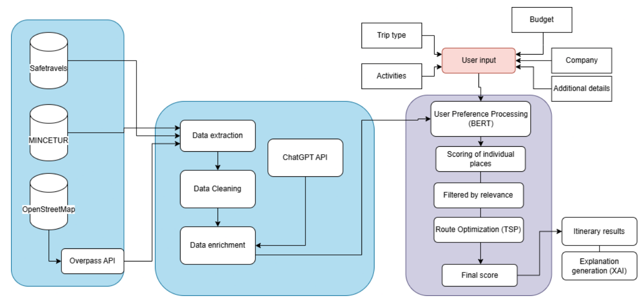

Fig. 3. Data integration flow from open sources and semantic processing.

aiming to achieve a clear separation between presentation logic, event handling, and data synchronization. This separation improves maintainability, scalability, and extensibility, facilitating future iterations and testing. Fig. 3 shows data integration flow from open sources and semantic processing.

Fig. 4 depicts the vertical architecture of the frontend. At the top is the Controller, responsible for handling user interaction events and sending HTTP requests to the backend. The middle section contains the Model, which temporarily stores client-side information, synchronizes data received from the backend, and maintains the interface state. At the bottom are the Views, responsible for presenting data clearly and dynamically to the user.

The interaction flow is bidirectional:

- The user interacts with the views, for example, by defining travel preferences or rating an itinerary.
- These actions are captured by the Controller, which sends HTTP requests to the backend through REST endpoints.
- The response is received by the Model, which updates its state and re-renders the views.
- Finally, the user visualizes the updated information without reloading the application.

The interface is designed to provide progressive navigation, guiding the user from preference definition to the exploration of specific itineraries.

The main views are:

- Preferences Screen: The first interaction stage. It allows users to select travel type (adventure, cultural, family, etc.), activities of interest, budget, and travel companions. This information serves as input for the recommendation engine.

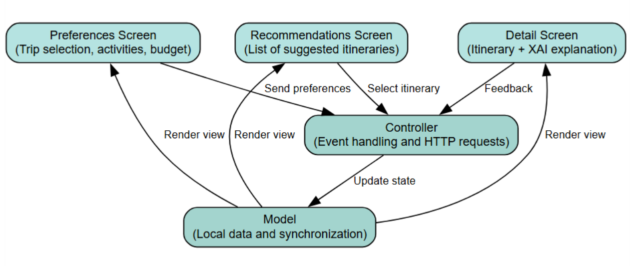

Fig. 4. Vertical frontend architecture based on the MVC pattern.

- Recommendations Screen: Displays a list of suggested itineraries generated by the hybrid recommendation model (BERT + collaborative filtering). Each itinerary includes a name, image, list of destinations, and a relevance score. The user can scroll through options and select the most suitable one.
- Itinerary Details Screen: Shows detailed information about the selected itinerary, including destination descriptions, suggested activities, and explainable AI (XAI) reasoning for the recommendation. Users can also rate the recommendation with stars, providing feedback that feeds the collaborative filtering module.

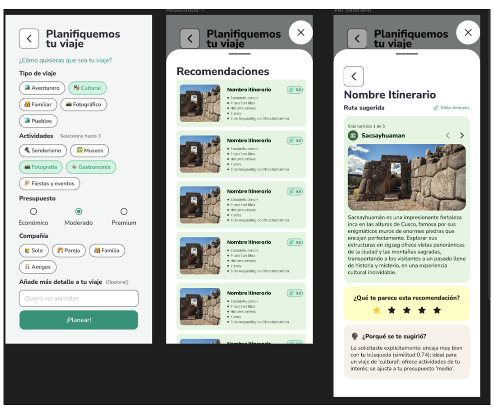

Fig. 5. Application screens.

The implementation coherently integrates all system components: a backend structured under Clean Architecture principles, a hybrid recommendation engine based on BERT semantic embeddings, a robust open-data integration pipeline, and a frontend organized under MVC. The coordination among these modules through a clear operational flow ensures efficient, modular, and scalable processing. This solid technical foundation enables the system not only to generate personalized and relevant recommendations but also to evolve easily to incorporate new functionalities, data sources, or recommendation methods in the future.

V. Experimentation

The experimentation phase aimed to evaluate the performance of the sustainable tourism recommendation system by applying the scenarios defined in the evaluation stage of the DSR methodology. Different types of tourist profiles representative of visitors in the Cusco region, Peru, were simulated.

A. Profiles and Evaluation Scenarios

Three main profiles were defined:

- Cultural Profile: focused on history, archaeological heritage, and visits to emblematic sites.
- Gastronomic Profile: oriented toward local culinary experiences and traditional markets.
- Adventure Profile: related to outdoor activities, trekking routes, and contact with nature.

Each profile was tested with different combinations of variables such as type of trip, budget, companionship, and preferred activities, thus generating multiple test scenarios. The scenarios used were:

This scenario structure allows for covering different types of tourist experiences and evaluating the system's ability to provide relevant and adaptive recommendations for various user profiles.

B. Test Execution

The evaluation was carried out by executing complete flows in the application for each scenario:

- A scenario with its defined variables is selected.
- The frontend sends the request to the backend.
- The BERT-based recommendation engine processes the inputs and computes the cosine similarity with available tourist destinations in the database.
- The system returns itineraries ranked by relevance.
- The average response time is recorded, and external feedback on the perceived usefulness of the recommendations is collected.

C. Evaluation Metrics

The following metrics were used to assess system performance:

- Cosine Similarity: measures the semantic relevance between the entered preferences and the generated recommendations.
- Average Response Time: evaluates the efficiency of request processing.
- External Feedback: validates the perceived usefulness of the recommendations from the user's point of view.

These metrics measure not only the technical precision of the system but also its performance and usability from a practical perspective. The quantitative results obtained are presented in the following Results section.

VI. Results

The proposal was evaluated based on the six previously defined scenarios (see Table II), representing different user profiles, travel types, budgets, and companionships. The analysis focused on three main dimensions:

TABLE II. SCENARIOS AND USER CHARACTERISTICS FOR THE RECOMMENDATION SYSTEM

Scenario	Profile	Budget	Companionship	Brief Description
E1	Cultural	Moderate	Couple	Heritage and historical visits.
E2	Cultural	Premium	Solo	Personalized individual experience.
E3	Gastronomic	Economic	Solo	Exploration of markets and restaurants.
E4	Gastronomic	Moderate	Friends	Group tours with tastings.
E5	Adventure	Premium	Friends	Mountain and trekking activities.
E6	Adventure	Economic	Family	Accessible group experience.

- Technical Performance: measured through the average cosine similarity, which evaluates the semantic correspondence between user preferences and the descriptions of the recommended tourist destinations.
- Performance: measured by the system's average response time per request, considering the full interaction from user input to the recommended itinerary.
- Perceived Usability: assessed through simulated external feedback, reflecting overall satisfaction with the obtained recommendations.

Cosine similarity values ranging from 0.80 to 0.85 show that the system achieves a high semantic correspondence between input preferences and generated recommendations. The average response time remained between 2.0 and 2.2 seconds, confirming stable and adequate performance for an interactive environment. Likewise, positive feedback —between 85% and 92%— indicates high acceptance of the suggested itineraries.

A. Complementary Metrics Suggested by Literature

While this work prioritizes evaluation through cosine similarity, response time, and external feedback, scientific literature on AI-based and BERT-based tourism recommender systems suggests complementing these measurements with broader indicators. These indicators allow for a comprehensive evaluation, considering not only technical quality but also acceptance and effectiveness from the end user's perspective.

The incorporation of these metrics would allow, for example:

- Objective comparison of the system's performance against existing solutions.
- Evaluation not only of semantic accuracy but also of user satisfaction and software quality.
- Identification of weaknesses in specific scenarios, such as recommendation errors (RMSE) or coverage biases (F1-score).
- Alignment of evaluation with international standards, facilitating publication and experiment replicability.

Specifically, MAE and RMSE are relevant when users rate recommended destinations, while F1-score and Accuracy are useful when classifying intentions or activity types. Retention and ISO 9126 extend evaluation toward user experience and software quality, complementing purely technical assessment.

B. General Interpretation

The results indicate that:

- The **BERT**-based semantic engine effectively captures user preferences in tourism contexts.
- The stable response time ($\sim$ 2 s) enables smooth interaction, suitable for mobile applications.
- The high positive feedback suggests that the recommended itineraries are relevant and satisfactory for different profiles.
- Incorporating recognized technical and usability metrics provides robustness and comparability in evaluation.

These findings demonstrate the technical and functional feasibility of the proposal, establishing a solid framework for future improvements, including more complex metrics, real user testing, and longitudinal evaluation. Table III shows average results per scenario.

VII. DISCUSSION

The proposed sustainable tourism recommendation system in Peru, integrating a BERT-based semantic model and a decoupled architecture through Clean Architecture, offers significant advances over traditional approaches, both algorithmically and structurally. This section discusses the advantages and challenges of this proposal compared to other approaches documented in academic literature, while integrating the experimental results.

A. Advantages over Traditional Approaches

Regarding recommendation models, BERT represents a substantial improvement over classical methods such as TF-IDF, Support Vector Machines (SVM), or Collaborative Filtering (CF). Traditional content-based approaches like TF-IDF rely on term frequency and lack semantic understanding, limiting their ability to capture user intent when rich descriptions or synonyms are used. SVM can classify tourism texts with moderate efficiency but requires significant feature engineering and does not model deep contextual relationships. Collaborative Filtering, widely used in early stages of digital tourism, faces well-known challenges such as cold start and poor interpretability of novel preferences, as noted by Uribe Iraola et al. (2023) [15].

In contrast, the BERT model implemented in this research, combined with cosine similarity measures, achieved an average similarity score of 0.83 across six evaluation scenarios, demonstrating a strong ability to capture semantic relationships

TABLE III. AVERAGE RESULTS PER SCENARIO DEFINED IN TABLE II

Scenario	Average Cosine Similarity	Response Time (s)	Positive Feedback (%)
E1 — Cultural (Moderate, Couple)	0.83	2.1	92
E2 — Cultural (Premium, Solo)	0.85	2.0	90
E3 — Gastronomic (Economic, Solo)	0.80	2.0	85
E4 — Gastronomic (Moderate, Friends)	0.83	2.1	87
E5 — Adventure (Premium, Friends)	0.84	2.2	88
E6 — Adventure (Economic, Family)	0.81	2.2	86

TABLE IV. COMPLEMENTARY METRICS COMMONLY USED IN THE LITERATURE TO EVALUATE TOURISM RECOMMENDER SYSTEMS

Metric	Type	Main Purpose	Reference
MAE (Mean Absolute Error)	Technical	Measures the average error between actual user ratings and system predictions. A low value indicates estimates close to real perception.	[6]
RMSE (Root Mean Square Error)	Technical	Penalizes large errors; useful for detecting recommendations poorly aligned with user interests.	[6]
F1-score	Technical	Combines precision and recall; ideal for evaluating models that classify user intentions or categories.	[12]
Accuracy	Technical	Proportion of correct recommendations relative to the total.	[12]
Retention	Usability	Measures the percentage of users who continue using the app; indicates a positive experience.	[15]
ISO 9126	Usability	Evaluates software quality attributes (usability, functionality, efficiency, maintainability).	[2]

between user profiles and sustainable destination descriptions. This result is consistent with the values reported by Hartatik et al. (2024), who showed that BERT outperforms classical methods, achieving an MAE of 0.1162 and RMSE of 0.1214. Cosine similarity, by measuring the angular distance between vectors in high-dimensional spaces, enabled precise recommendations without the need for additional supervised training. However, a challenge identified was the higher computational cost compared to classical approaches, which may limit deployment on resource-constrained devices.

B. Architectural and Technical Contributions

From a structural design standpoint, adopting Clean Architecture as the system's backend foundation was a key decision compared to microservices approaches commonly used in large-scale recommender systems. While microservices offer high scalability and distributed deployment, studies such as Wijayanto Sejati (2023) indicate that in medium-scale mobile applications, Clean Architecture provides greater clarity in separation of concerns, maintainability, and testability [19]. In this project, this structure effectively decoupled the domain layer—hosting the recommendation engine—from open data sources (MINCETUR, SafeTravels, and OpenStreetMap), achieving a more stable architecture that can evolve without complex infrastructure dependencies.

In the Peruvian context, where sustainable tourism is not yet widely digitized, this architectural decision proved practical. During implementation, it allowed for seamless integration of recommendation and data management modules without affecting the presentation layer (based on MVC). This addresses a need observed in the literature: many reviewed applications implement only presentation patterns, lacking a clearly defined domain layer, as noted by Villacorta et al. (2022) and Ccuno Carlos et al. (2022) [17], [2]. Combining MVC in the frontend and Clean Architecture in the backend provides a balance between simplicity and technical robustness.

C. Evaluation and User Experience

From an evaluation perspective, the metrics most used in similar studies—MAE, RMSE, F1-score, and accuracy—support the effectiveness of BERT models in analyzing unstructured text and implicit preferences. In this study, the average similarity of 0.83 confirms that the proposal effectively aligns semantic destination descriptions with user profiles. Additionally, literature emphasizes that technical precision should be complemented with experience and retention indicators, such as the 98.35% retention reported by Uribe Iraola et al. (2023). This opens the possibility of incorporating mixed evaluation in future versions, combining technical metrics with sustainability perception and local tourism impact (see Table IV).

D. Limitations and Improvement Opportunities

Nevertheless, some important limitations must be acknowledged. Running BERT requires minimum computing infrastructure and stable connectivity, which can pose a barrier in rural areas or low-end devices. Moreover, integrating open data from Peru's tourism ecosystem presented challenges in normalization, consistency, and semantic updating. This suggests that future iterations could benefit from lighter models such as DistilBERT, knowledge distillation techniques, or edge computing mechanisms to reduce load on the central server.

E. Final Synthesis

In summary, the discussion shows that the combination of BERT with cosine similarity and a Clean Architecture represents a solid solution for contextualized personalization in sustainable tourism, validated both by the experimental results (similarity score of 0.83) and by evidence reported in the literature. At the same time, it highlights real challenges in optimization and scalability that represent opportunities for improvement in future development phases. The results suggest that tourism digitalization policies should prioritize semantic models and modular interoperable architectures.

VIII. Conclusions and Future Work

This research proposed and implemented a sustainable tourism recommendation system in Peru that combines a semantic model based on BERT with a decoupled architecture using Clean Architecture for the backend and the MVC pattern for the frontend. This integration effectively addressed the personalization of tourist itineraries, improving the relevance and contextualization of recommendations compared to traditional methods.

The experimental results showed that the model achieved an average similarity of 0.83 between user preferences and the descriptions of sustainable tourist destinations, demonstrating its ability to capture complex semantic relationships and generate suggestions aligned with users' real interests. Additionally, stable response times between 2.0 and 2.2 seconds were achieved, suitable for mobile applications. External feedback obtained through simulated scenarios also validated the practical relevance of the recommendations, confirming that the system not only provides technical precision but also perceived value for the user.

From an architectural standpoint, the use of Clean Architecture proved to be an efficient alternative to more complex architectures such as microservices, as it promotes maintainability, separation of concerns, and ease of system evolution. This approach enabled the modular integration of open data sources from MINCETUR, SafeTravels, and OpenStreetMap without compromising performance or application stability.

Methodologically, the adoption of the Design Science Research approach facilitated a structured and rigorous development process, ensuring traceability between problem identification, design, implementation, evaluation, and communication of results. This guarantees that the solution is not only technically viable but also scientifically grounded.

A. Future Work

Despite the progress achieved, there are opportunities for improvement and expansion that could enhance the impact of the proposal:

1) Model optimization: It is recommended to explore lightweight variants of BERT, such as DistilBERT, or knowledge distillation techniques that reduce computational load and allow greater accessibility on mid- and low-end mobile devices.

2) Geographical scalability: Expand the coverage of tourist destinations to other regions of the country by incorporating additional and updated open data sources.

3) Evaluation in real scenarios: Conduct tests with end users in real environments to complement technical metrics with indicators of usability, satisfaction, and perception of sustainable tourism.

4) Architectural improvements: Integrate edge computing or local caching mechanisms to further optimize response times and offline availability.

5) Incorporation of social and sustainability metrics: Explore indicators of local impact, tourism footprint, or sustainability to align recommendations with public policies and sustainable development goals.

In conclusion, the research demonstrates that the combination of semantic models based on BERT and Clean Architectures constitutes a solid and adaptable solution for promoting sustainable tourist destinations in Peru. The experimental validation, together with the literature review, confirms that this approach has the potential to significantly contribute to tourism diversification, decentralization of visitor flows, and the strengthening of local development.

References

[1] Alamsyah, D. P., et al. (2024). Sustainable Tourism Recommender Systems: A Systematic Literature Review. Sustainability, 16(11), 4532.

[2] Ccuno Carlos, P., Chura Chambi, P., Lipa Ochoa, J., & Sulla-Torres, J. (2022). Development of a Mobile Application to Provide Employment Information to Private-Sector Workers in Peru.

[3] Daries, J., Jaime, V., & Bucaram, S. (2021). Evolución del turismo en Perú 2010–2020, la influencia del COVID-19 y recomendaciones pos-COVID-19. Banco Interamericano de Desarrollo.

[4] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[5] Gregor, S., & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. MIS Quarterly.

[6] Hartatik, H., Isnanto, R. R., & Warsito, B. (2024). Tourist Recommendations Based on Online Reviews with Sentiment Analysis using BERT in Central Java Tourism.

[7] Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. MIS Quarterly.

[8] Lin, C., & Nuha, U. (2023). Sentiment analysis of Indonesian datasets based on a hybrid deep-learning strategy. Journal of Big Data, 10.

[9] MINCETUR. (2023). Reporte regional de turismo 2023. Gobierno del Perú.

[10] MINCETUR. (2024). Reporte regional de turismo 2024. Gobierno del Perú.

[11] Martin, R. C. (2017). Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall.

[12] Nguyen, T. H., Tran, Q. T., & Le, A. V. (2023). AI Chatbot for Tourist Recommendations: A Case Study in Vietnam.

[13] Nguyen, T. H., Tran, Q. T., & Le, A. V. (2024). BERT for Tourism Recommendation in Vietnam. Multimedia Tools and Applications.

[14] Rodríguez, G., Wong, L., & Mauricio, D. (2017). A systematic literature review about software requirements elicitation. Journal of Engineering Science and Technology.

[15] Uribe Iraola, B. M., Salcedo Vaccaro, D. E., Aures García, A. A., & Flores Moroco, J. A. (2023). Mobile App to Improve the Tourist Experience by Using Multilayer Perceptron Network and Collaborative Filtering.

[16] Van Der Wal, P., Strisciuglio, N., Azzopardi, G., & Mocanu, D. (2025). Multilayer perceptron ensembles in a truly sparse training context. Neural Computing and Applications.

[17] Villacorta, R. D. M., Aguero, C. O., & Laberiano-Andrade, A. (2022). Implementation of a Mobile Application Oriented to the Tourism Sector.

[18] vom Brocke, J., Hevner, A., Maedche, A., & Winter, R. (2020). Special Issue: Accumulation and Evolution of Design Knowledge in Design Science Research. Journal of the Association for Information Systems.

[19] Wijayanto Sejati. (2023). Implementing Flutter Clean Architecture for Mobile Tourism Application Development.

[20] Xiao, L., Li, Q., Shen, J., Yang, Y., & Li, D. (2024). Text classification algorithm of tourist attractions subcategories with modified TF-IDF and Word2Vec. PLOS ONE, 19.

[21] Yang, X., Ali, N., & Tat, H. (2025). Study on Tourism Development Using CRITIC Method for Tourist Satisfaction. IEEE Access, 13, 79915-79928.