Design of a Music Store Application (Ortizo)

Esteban Alejandro Villalba Delgadillo
Santiago Marin Paez

Introduction

Developing virtual stores for product purchases is a common solution in e-commerce, but effective inventory management and automatic receipt generation remains a challenge. Previous solutions have tried to optimize these functions, but lack scalability and flexibility in data handling and future updates. The challenges include integrating multiple modules (products, users, receipts) and efficient real-time data management while maintaining information security.

Goal

The main goal is to create a virtual store that allows users to buy musical instruments and accessories, while the system manages inventory and generates receipts automatically. The expected end product is a functional shop with a relational database capable of supporting future modules such as promotions and a shopping cart.

Proposed Solution

The proposed solution is based on a client-server architecture with a relational database supporting the main operations: inventory management, users and receipts. The database design follows a systematic ten-step approach, where key components, entities and attributes are defined. Principal entities include instruments, accessories, categories, brands, inventory, receipts and users. The relationships between these entities ensure consistent data handling, maintaining referential integrity and system scalability.

Results

The results include the successful design of the virtual store, with a database that allows automatic receipt creation, real-time inventory management and integration of new modules in the future. The following is a comparative table between the features of the developed virtual store and previous solutions:

Feature	Previous S.	After developmetn
Inventory management	Daily update	Real time update
Generation of receipts	Manual	Automatic
Ease of shopping	Area limitation	No Limitation to the area

Conclusion

The project achieves its goal by designing a functional virtual store, backed up by a relational database. The research question was answered by creating a scalable and flexible architecture that allows future integration of new features without compromising system performance.