



# Simplified Implementation of Microservices for Banking Management

**David Martinez, Julian David Pulido**

Universidad Distrital Francisco Jose de Caldas  
Dmartinezr@udistrital.edu.co, judpulidoc@udistrital.edu.co



## ABSTRACT

The problem of user and transaction management in a digital bank presents significant challenges related to the implementation of a scalable and accessible architecture for intermediate-level developers. In this work, a solution is proposed based on the simplification of the microservices architecture, dividing it into two main services: one for user management and another for transaction management. The results show that the proposed solution reduces the implementation complexity and improves the efficiency of the system, achieving a high level of success in unit testing and integration, with a lower overhead compared to more distributed architectures.

## INTRODUCTION

### Introduction

In recent years, the digitalization of financial services has transformed the way people manage their money, facilitating fast and secure transactions without the need to visit a bank branch. However, this change has also brought with it significant challenges in implementing scalable architectures that can be easily managed, especially by developers with an intermediate level of experience. The need for a simplified microservices architecture for digital banking systems is crucial to ensure that these systems can be maintained and scaled without introducing unnecessary complexity.

Popular solutions such as Nequi and Movii have managed to offer a complete digital banking experience. These applications allow users to manage their finances, make transfers and payments, and even apply for loans, all from a mobile phone. Nequi, for example, uses a microservices-based architecture that is deployed across multiple servers, ensuring the availability and scalability of its services. However, this type of infrastructure comes with significant challenges, such as managing communication between microservices and the complexity of HTTP requests. Similarly, Movii has opted for an architecture that guarantees great flexibility, but requires a considerable level of technical expertise to be properly implemented and maintained.

In this work, we address the problem of creating a simplified microservices solution that allows developers with an intermediate level of expertise to efficiently implement a digital banking management system. Unlike the traditional architectures used by Nequi or Movii, our proposal is based on reducing complexity by dividing the system into just two main microservices: one for user management, which includes authentication and profile management, and another for transaction management, which covers accounts, transaction history, and sending notifications. This simplified approach minimizes the number of communication points, reducing the need to handle complex network protocols and decreasing the likelihood of errors.

In computational terms, a microservices-based architecture allows for clear separation of responsibilities, facilitates independent scalability of services, and promotes modular development. However, this modularity comes at the cost of a greater need for coordination between services, especially when dealing with services deployed in a distributed manner across multiple servers. To overcome these barriers, in our approach we have chosen to deploy the microservices on a single local server, which eliminates the need to handle HTTP requests and ensures direct communication between components. This makes the system more accessible to developers who are just starting out with microservices architecture, but still want to enjoy the benefits of a clear separation of responsibilities.

Referencing previous techniques, it is important to mention how current solutions apply microservices design patterns, such as asynchronous communication with message queues and the use of REST APIs for service exposure. While these approaches are useful for large-scale distributed systems, they also involve a significant learning curve and more advanced computational resources. In our work, we simplify these patterns by removing the distributed communication layer and implementing a modular monolithic architecture, which allows for a similar solution, but with a more affordable and straightforward approach.

This paper describes the problem context in detail, the proposed solution to overcome it, and the results obtained by applying this simplified microservices approach. The main motivation behind this research is to demonstrate that it is possible to implement an effective and scalable digital banking solution without having to resort to highly complex infrastructures that require deep knowledge in network management and distributed microservices.

## References

- [1] Castillo Uparela, D. E., & Galindo Monsalve, R. (2021). Concepción organizacional de Nequi y el modelo de negocio de banca digital en el contexto colombiano. *Revista de Tecnología Financiera*, [Enlace al artículo].

## Method and Materials

The design of the proposed solution is based on a simplified microservices architecture, aimed at managing users and transactions in a digital banking environment. This choice is based on the need to create a scalable system that allows easy administration and maintenance, without overloading the developer with unnecessary complexities. Instead of implementing multiple services dispersed across different servers, the solution is structured around two main microservices: the User Service, which handles authentication and profile management, and the Transaction Service, which handles accounts and transactions. This modular approach ensures that each microservice can be developed and scaled independently, facilitating future expansions of the system.

The decision to use a local server to deploy both microservices allows for efficient communication between them, avoiding the latency associated with HTTP requests in a distributed architecture. This is particularly beneficial for mid-level developers looking to simplify the implementation of a microservices system. By keeping internal communication on a single node, points of failure are minimized and system performance is improved. Additionally, the complexity of session management and authentication in distributed environments is avoided, which can be a significant challenge for those just starting to work with microservices architectures.

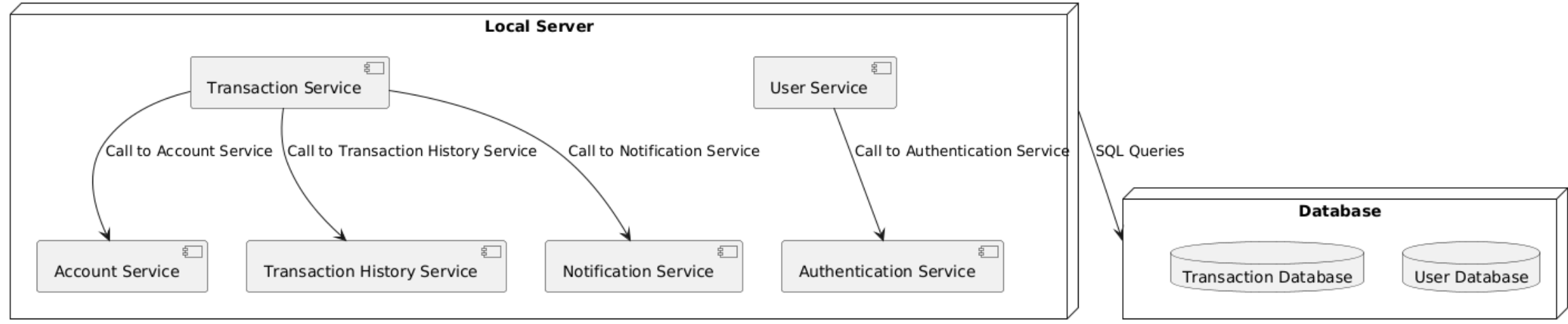


Figure 1: Diagrama de Despliegue del Sistema de Gestión Bancaria.

In terms of design, common design patterns in microservices architectures have been used, such

as the Single Service pattern, which allows each component to handle its own responsibilities. For example, the User Service not only handles authentication, but also manages user profile information, making it a central point for user-related operations. This not only makes code maintenance easier, but also allows developers to focus on specific functionalities without losing sight of the interconnection of services.

To illustrate this architecture, the class diagram is presented, showing the relationships between the different components of the system. This diagram highlights how the User, Profile, and Authentication classes are interrelated within the User Service, while the Account, Transaction, and Notification classes form the Transaction Service. Through this diagram, one can appreciate the clear separation of responsibilities that makes the system easier to understand.

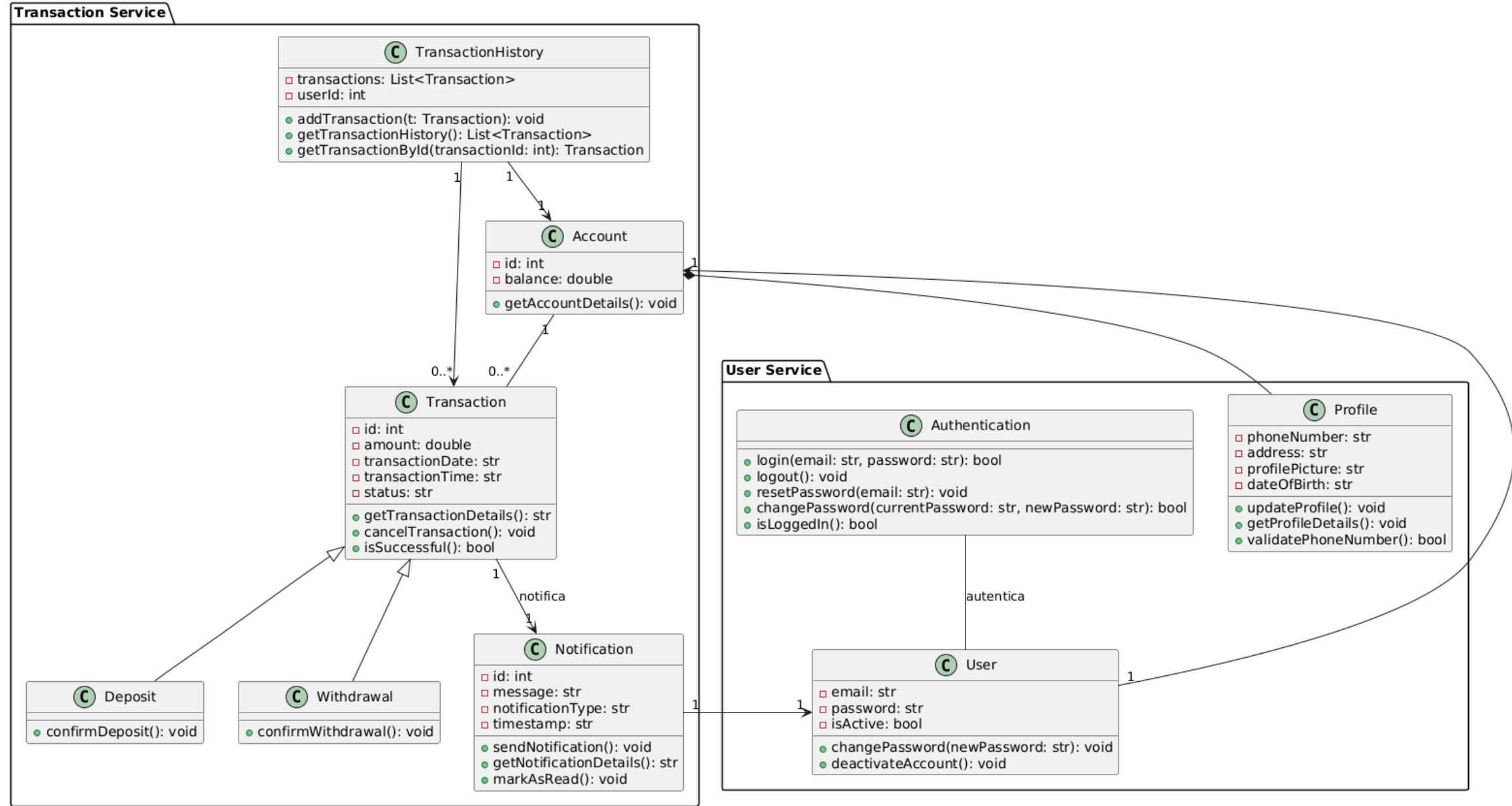


Figure 2: Diagrama de Clases del Sistema de Gestión Bancaria.

Additionally, an activity diagram is included that details the workflow in the system during a bank transfer. This diagram allows the visualization of the critical stages of the process, from login to success or error notification, providing a clear understanding of how the different components interact in real-world situations. The inclusion of these diagrams not only increases the explanatory power of the design, but also helps developers and stakeholders quickly grasp the structure and operation of the proposed system.

Finally, the use of a simplified approach to implementing microservices ensures that the system is accessible and manageable for intermediate-level developers, allowing them to gain experience in building modern solutions without being overwhelmed by the complexity of more advanced architectures. This makes the solution an effective tool for both bank management and for the learning and professional growth of the developers involved in the project.

## Results

Through the implementation of the simplified microservices architecture for user and transaction management in the digital banking system, the project is anticipated to achieve several key outcomes that will benefit end users. One of the main goals is to simplify the use of banking services, making them more accessible to people of various ages and levels of technological experience. The intuitive user interface and the elimination of the complexity associated with managing multiple external microservices should result in a smooth and satisfactory user experience.

As for the tests that are planned to be carried out, it is anticipated that unit tests will be carried out that will evaluate each individual component of the system. The philosophy behind these tests is to ensure that each microservice operates independently and meets the established functional requirements. At least 50 unit tests are planned to be carried out in total, covering all critical functionalities of the services. The expectation is to achieve a success rate of 95

In addition, integration tests will be carried out to ensure that the interaction between the microservices is carried out effectively. These tests are essential to confirm that communication and data flow between the User Service and the Transaction Service operate smoothly. It is anticipated that at least 10 integration tests will be performed, and a success rate of 90

Finally, acceptance tests are planned that will focus on evaluating how the system meets end-user expectations. These tests will be designed based on previously defined success criteria, ensuring that the solution not only functions correctly, but also provides a satisfactory user experience.

To facilitate understanding of these expected outcomes, a table will be provided that will summarize the test definitions and their objectives, as well as the desired success criteria. It is also planned to include graphs to compare the anticipated performance of the system with other digital banking solutions in terms of user satisfaction and operational effectiveness.

## Conclusions

In this work, a solution based on a simplified microservices architecture has been proposed for managing users and transactions in a digital banking system. Through the implementation of only two main microservices – the User Service and the Transaction Service – the aim is to reduce the complexity inherent in existing digital banking solutions. This simplification will allow mid-level developers to implement and maintain an effective system without the burden of managing multiple distributed services, facilitating access to banking services for a wider audience.

The expected results of this proposal not only include a more accessible and user-friendly system, but also a significant improvement in the user experience when making transactions. With a focus on unit testing and integration, a high level of quality and performance is anticipated, which will ensure that the system meets end-user expectations. By addressing the challenges of architectural complexity in digital banking, this solution represents a significant advance in the creation of modern and efficient financial applications.

## References

- [1] Castillo Uparela, D. E., & Galindo Monsalve, R. (2021). Concepción organizacional de Nequi y el modelo de negocio de banca digital en el contexto colombiano. *Revista de Tecnología Financiera*, [Enlace al artículo].
- [2] Flórez Aparicio, E. R. (2023). Impacto de las nuevas tecnologías en la utilización de los productos bancarios digitales tradicionales: caso Nequi y Daviplata. *Nombre de la Revista o Publicación*, [Enlace al artículo].