DEVELOPMENT OF A DIGITAL WALLET USING OBJECT-ORIENTED PROGRAMMING AND SOFTWARE DESIGN PATTERNS

Cristian Santiago López Cadena, Carlos Alberto Barriga Gámez
20222020027, 20222020179

INTRODUCTION

This project aims to develop a functional digital wallet platform using the object-oriented programming paradigm and the software patterns.
Object-oriented programming is defined as a programming paradigm that attempts to simulate things in the real world through elements called objects. These objects have some characteristics such as inheritance, polymorphism, encapsulation, and abstraction. Likewise, these objects are defined as a series of behaviors called methods and properties known as attributes.

According to the authors [4], design patterns are proven solutions to certain problems encountered when designing software.
On the other hand, the author [3] defines digital wallets as software that allows us to store electronic money for later use in online commerce. Likewise, the author states that digital wallets are used as a means of payment to send and receive money safely.

GOAL

Develop a digital wallet, using the object-oriented programming paradigm and the software design patterns, to obtain a scalable, maintainable and efficient application.

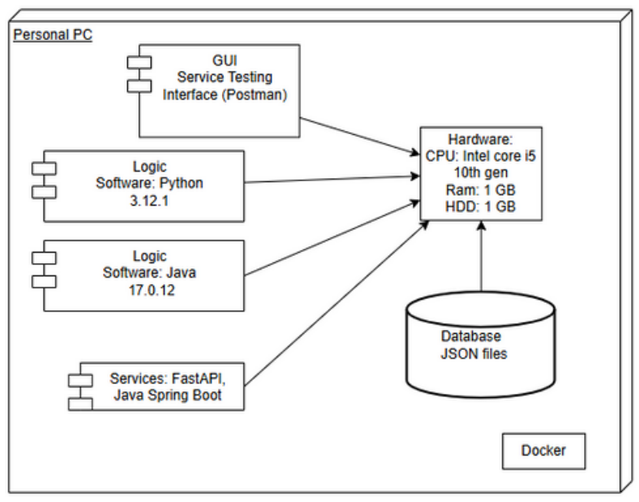
PROPOSED SOLUTION

Python version 3.12.1 and Java version 17.02.12 were used for the development of the logical part of the software, in the same way for the development of the data layer, it was decided to use Json files.
We use FastApi and Spring boot tools to manage and unify the services of the Python and Java backends.
The verification of the Java and python services is carried out using the Postman software.

RESULTS

The application design process was carried out, identifying user stories, creating class, activity, sequence, and deployment diagrams.
To verify the correct functioning of the software, it was decided to use user stories as a way of verifying the effectiveness of the software

The persistence of the information were tested by checking whether the information can be stored and manipulated correctly in the json files.
In the same way, tests were carried out on the application services in order to determine whether they generate the requested information in each case.
We performed a detailed analysis of the design patterns we implemented, verifying to implement only those hat really added value to the project
Tools such as pylint and black formatter were also used to verify and avoid code smell errors in the project



Deployment diagram

CONCLUSION

The development of a digital wallet using object-oriented programming and design patterns enabled a modular, scalable, and maintainable system.

SOLID principles reduced dependencies, enhancing code organization and reusability. The project integrated Python and Java backends with FastAPI and Spring Boot for communication, leveraging each technology's.

The combination of Python and Java as backends, along with the use of FastAPI and Spring Boot for communication between the two, allow us to create a scalable, modular and maintainable system.
Data persistence with JSON files proved an efficient and easy-to-use solution for storing information, although in higher scalability scenarios it would be advisable to consider more robust databases.
Tests performed with Postman and code analysis tools such as pylint and black formatter ensured the quality and functionality of the software.
Finally
Finally, through the review of the user stories, it was possible to verify that the system met the requirements set out in the development of the project.

BIBLIOGRAPHY

[1] S. Valbuena and S. A. Cardona, "Object-oriented programming principles" Elizcom S.A.S, 2018, pp. 7.
[2] C. A. Robleto, "Electronic Commerce: Background, Definitions and Subjects", 2004, pp. 6-8.
[3] G.Bellindo, "Electronic wallets: a tool for entrepreneurship in the digital age", 2023
[4] E.Gamma, R.Helm, R. Johnson, J.Vissides, "Design patterns. Elements of reusable object-oriented Software", 1994

[5] Autentia, "Software design, principles and patterns of the software development", 2012, pp. 9-17.
[6] Amazon, "What is the difference between monolithic and microservices architecture?", 2023.