UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN - DCCO**

**CARRERA DE INGENIERÍA DE SOFTWARE**

PERIODO : noviembre 2024 – marzo 2025

ASIGNATURA : Desarrollo web Avanzado.

NOMBRES : Kevin David Asmal Haro

NIVEL-PARALELO : Quinto / 2296.

FECHA DE ENTREGA: 06 de enero de 2025.

**Introduction.**

This document examines the essential ideas of SOAP, REST, and the Microservices Architecture Pattern, which are vital for creating and building contemporary software systems. It covers their purposes, main features, significant distinctions, and real-world applications in business software. The arrangement of the material is organized and thorough to offer a clear comprehension and useful advice for effective use.

**Objective Systems.**

* General Objective.

To examine the ideas, goals, and real-world uses of SOAP, REST, and the Microservices Architecture Pattern, focusing on their advantages, drawbacks, and optimal methods for successfully applying them in business systems.

* Specific Objectives.

To describe the main characteristics and applications of SOAP and REST as web service frameworks.

To discuss the Microservices Architecture Pattern, highlighting its benefits and difficulties in software development.

To give real-life examples of how these technologies can be applied in business situations.

To provide practical advice on choosing the right method based on the specific needs of a project.

**Development.**

**SOAP (Simple Object Access Protocol).**

Is an XML-based protocol created for organized communication between applications across a network. It offers a consistent method to describe, find, and connect web services.

* Key Features:

Defines services using WSDL (Web Services Description Language).

Utilizes standard transport protocols, usually HTTP or SMTP.

Provides enhanced security features through WS-Security.

* Advantages:

Compatible and dependable for large-scale business applications.

Facilitates complicated operations and transactions across different systems.

Communication is platform-independent thanks to its XML format.

* Limitations:

More complex than REST.

Development and upkeep can be challenging due to XML data formats.

* Use Cases:

Financial applications that need secure and transactional processes.

Government services requiring strict adherence to regulations.

Diagram Suggestion: A visual illustration of the SOAP request and response process.

**REST (Representational State Transfer).**

Is a design approach for creating applications that connect over a network. It makes communication easier by using the common HTTP protocol.

* Main Features:

Operations that do not maintain state to improve scalability.

Compatible with various data formats including JSON, XML, and plain text.

Employs common HTTP methods: GET, POST, PUT, DELETE.

* Benefits:

More lightweight and quicker than SOAP.

Simple to incorporate into contemporary web and mobile apps.

Provides flexibility for programmers due to its straightforwardness and adaptability.

* Drawbacks:

The absence of rigid standards may cause inconsistencies across different implementations.

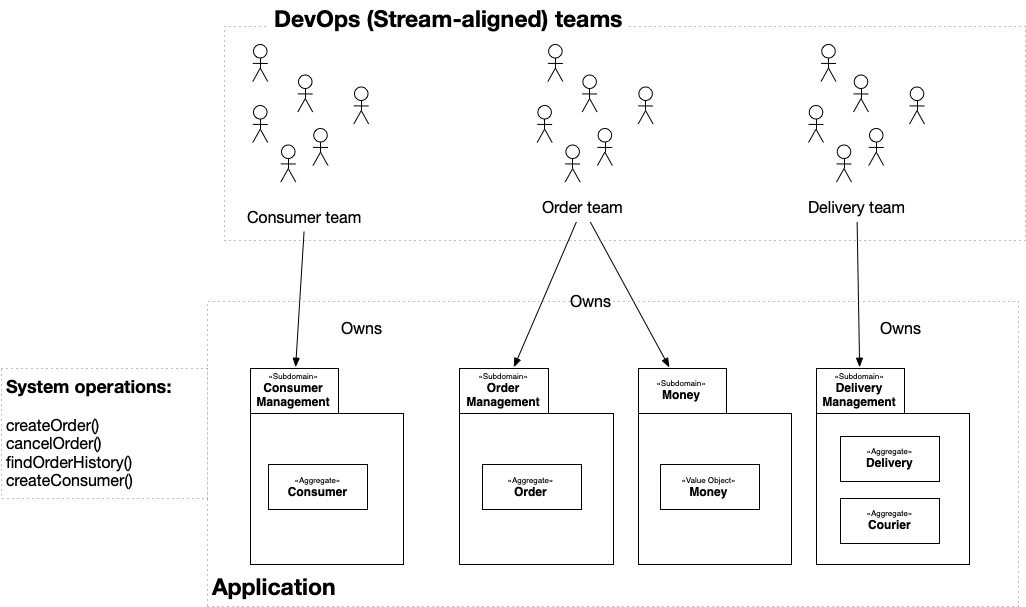
Might not be suitable for systems requiring high security or heavy transactions.

* Examples of Use:

Social media sites and online shopping platforms.

**Microservices Architecture Pattern**

Microservices Architecture is a method for building software applications as a collection of small, self-sufficient services that interact through APIs.



* Main Features:

Every microservice corresponds to a specific business operation.

Independent control over development, deployment, and scaling.

Interaction occurs through simple protocols like REST or messaging platforms.

* Benefits:

Scalability: Each service can grow on its own.

Flexibility: Different technologies and programming languages can be used for each service.

Better fault isolation: Issues in one service won't affect the whole system.

* Drawbacks:

Greater complexity in managing distributed systems.

Demands strong tools for monitoring, logging, and deployment.

* Examples of Use:

Companies such as Netflix, Amazon, and Uber rely on microservices to manage large-scale, high-traffic applications.

Diagram Idea: An architecture diagram illustrating microservices, an API Gateway, databases, and communication between services.



**Conclusions**.

SOAP is still a popular option for applications that need strong security and reliable transactions.

REST has transformed web services because of its ease of use, adaptability, and efficiency.

The Microservices Architecture Pattern provides exceptional scalability and modularity for big applications, but it demands significant expertise and the right tools for effective management.

Choosing between SOAP, REST, or Microservices should depend on the particular requirements of the project and the resources at hand.

**Bibliography.**

Chakray. (2016, diciembre 21). *¿Qué diferencias hay entre REST y SOAP y cuál es mejor?* Chakray. https://www.chakray.com/es/que-diferencias-hay-entre-rest-y-soap/

*REST versus SOAP: diferencias*. (s/f). Redhat.com. Recuperado el 6 de enero de 2025, de https://www.redhat.com/es/topics/integration/whats-the-difference-between-soap-rest

(S/f). Amazon.com. Recuperado el 6 de enero de 2025, de https://aws.amazon.com/es/compare/the-difference-between-soap-rest/

*Microservices Pattern: Microservice Architecture pattern*. (s/f). microservices.io; Chris Richardson. Recuperado el 6 de enero de 2025, de https://microservices.io/patterns/microservices.html