MATTER	Advanced Web Programming	**NRC**	8011
RACE	Software Engineering	**Teacher**	Dr. Edison Lascano
THEME	Rest		
Name	Yeshua Amador Chiliquinga Amaya		

1. Introduction:

REST (Representational State Transfer) is a style of software architecture designed for communication between systems, especially in distributed environments such as the web. Proposed by Roy Fielding in his doctoral dissertation in 2000, REST is based on simple principles and web standards such as HTTP. Its main objective is to facilitate interoperability between heterogeneous systems through uniform interfaces and well-defined resources.

2. Research

Rest would be defined by the following terms:

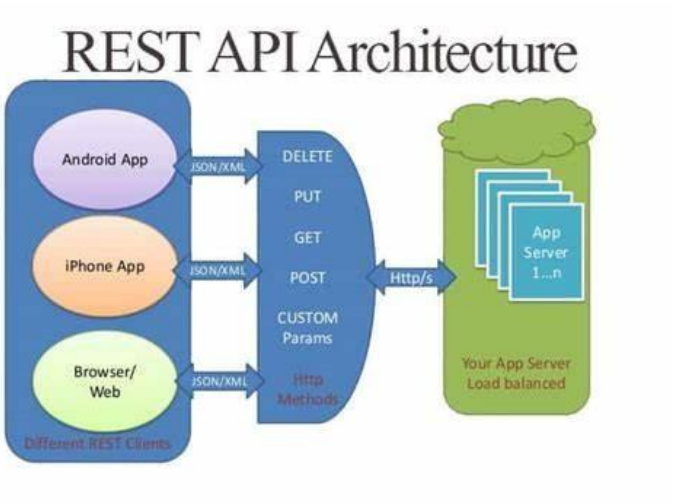
- Simplicity in design and adoption thanks to web standards.
- Platform and programming language independence.
- Scalability and flexibility in distributed systems.

REST is not a protocol, but a set of architectural constraints:

- **Client-server:** Separates the user interface from the data warehouse.
- **Stateless:** Each client request to the server must contain all the information needed to process it.
- **Cache:** Responses should indicate whether they are cacheable to improve performance.
- **Uniform interface:** Establishes a standard way to interact with resources.
- **Tiered system:** The client does not interact directly with the internal layers of the server.
- **Code on demand (optional):** The server can send executable code to the client.

Using HTTP in REST:

- The most common HTTP methods in REST are:
  - **GET:** Retrieve a resource.
  - **POST:** Create a new resource.
  - **PUT:** To update or replace an existing resource.
  - **DELETE:** Delete a resource.



3. Analysis

Modern REST implementation uses specific HTTP verbs (GET, POST, PUT, DELETE) for CRUD operations, providing an intuitive and predictable interface [2]. Endpoints follow a hierarchical structure of resources, and HTTP status codes effectively communicate the outcome of each operation. The clear separation between client and server allows both to evolve independently.

4. Conclusion

REST continues to be fundamental in modern web development, especially in microservices architectures. Its simplicity and scalability make it ideal for public APIs and distributed systems

5. References

Fielding, Roy. (2000). "Architectural Styles and the Design of Network-based Software Architectures"

Richardson, Leonard. (2022). "RESTful Web APIs"