

SYSTEMS ENGINEERING

Enterprise Architectures

Laboratory 4

Luis Daniel Benavides Navarro

Author:
Angie Daniela Ruiz Alfonso

Contents

1	Introduction	2
2	Architecture	2
3	Results	4
4	References	5

1 Introduction

In this laboratory, I built a web server (Apache type, which can respond to multiple non-concurrent requests.) In Java, the server can deliver html pages and PNG type images, and use the IoC framework to build web applications. From POJOS. Using the above content, I built a sample web application and deployed it on Heroku.

2 Architecture

The architecture used is the Client-server and we can see the physical design in the figure 1. This architecture consist in a computer network, in which many clients (remote processors) request and receive services from a centralized server (host). The client computer provides an interface to allow computer users to request the server's services and display the results returned by the server. The server waits for requests from clients and then responds to them. Ideally, the server provides a standardized transparent interface to the client, so that the client does not need to know the details (ie, hardware and software) of the system that is providing the service. Clients are usually located on workstations or personal computers, while servers are usually located elsewhere on the network on more powerful computers.

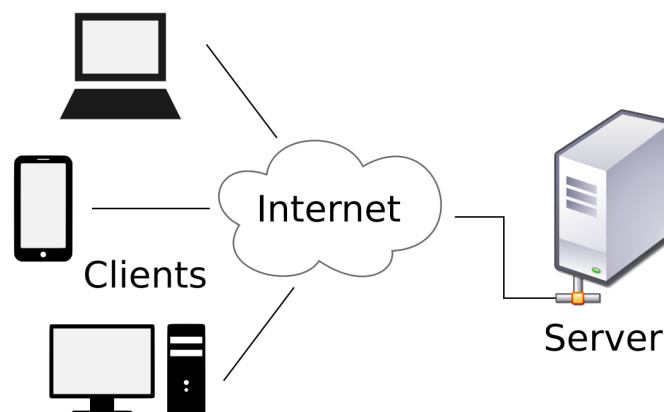


Figure 1: The architecture used

This calculation model is particularly effective when the client and the server each perform routine tasks. For example, in hospital data processing, the client computer may be running an application program for inputting patient information, while the server computer is running another program that manages a database that permanently stores information. Many clients can access server information at the same time, and client computers can perform other tasks at the same time, such as sending emails. Because both client computers and server computers are considered smart devices, the client-server model is completely different from the old "mainframe" model. In the old "mainframe" model, the centralized mainframe is its associated "dumb" model. "The terminal performs all tasks.[1]"

Inversion of Control (IoC or IOC) describes a system that follows the following principle: the control flow within an application is not controlled by the application itself, but by the underlying framework. Usually, in such an architecture, when writing an application, it should be bound to the application framework by handling framework events or inserting framework extension points.²

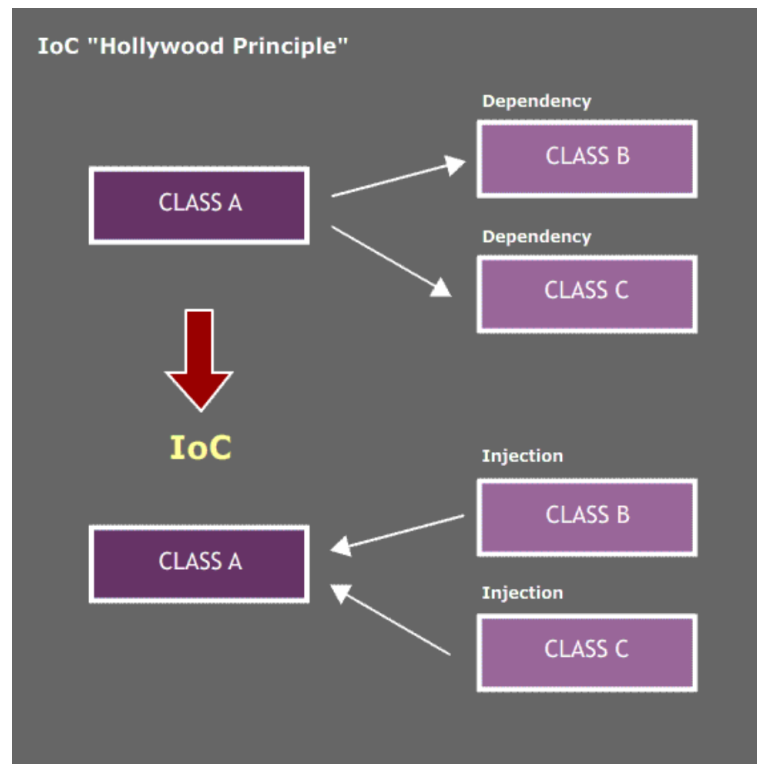


Figure 2: The framework used

The IOC container, also known as the dependency inversion (DI) container, is a factory designed to facilitate dependency injection.

3 Results

The results that were obtained in this laboratory are in the figure 3.

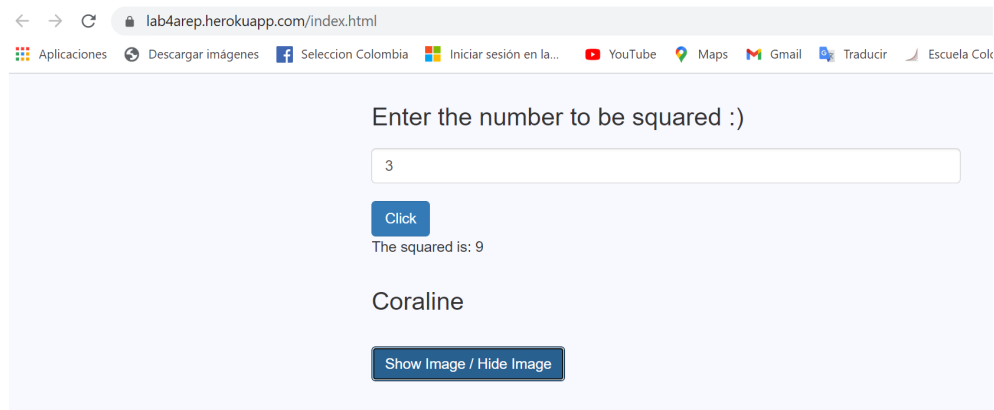


Figure 3: The results obtained

4 References

- [1] DevIQ., *Inversion of control*, <https://deviq.com/principles/inversion-of-control>, Accessed on 2021-02-22.