

# Enterprise Architecture Project

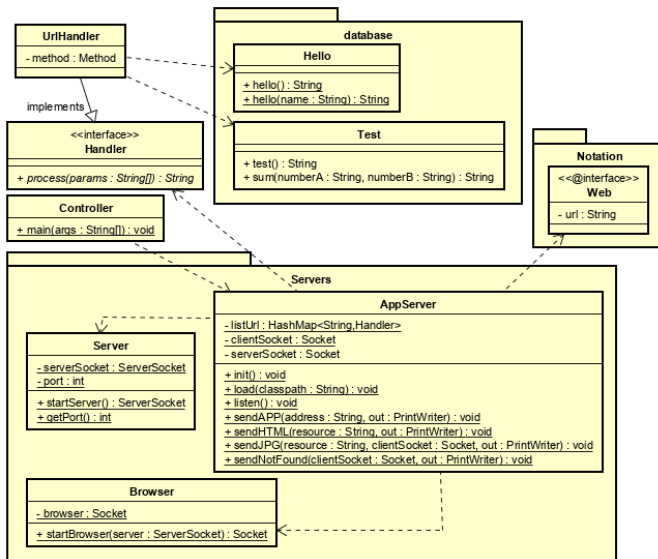
Juan Camilo Velandia Botello

**Abstract—** In this document there is the specification about the first enterprise architecture project.

## I. INTRODUCTION

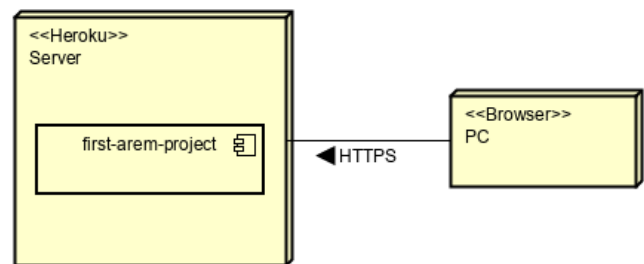
THIS project is about the construction of a web server (Apache type) in Java. The server is capable of sending HTML pages and JGP images, and it is also capable of searching for requested resources such as java classes and receiving parameters. In addition, this server provides an IoC framework for building web applications from POJOs (Plain Old Java Objects).

## II. SOFTWARE ARCHITECTURE



This architecture was modeled according to the requirements established in the project. It seeks to generate stability and scalability in the project.

## III. DEPLOYMENT ARCHITECTURE



The web application is deployed in Heroku. There are two nodes. The browser and the Heroku server, where our web application is located. The browser requests a resource by https to our application, and then the server answers the request to the browser.

## IV. WEB APPLICATION DESCRIPTION

This application has a controller which starts the AppServer. The AppServer class has a list of URLs containing hello and test classes. These classes were made so that users can search for them in the browser. Also, the AppServer class begins a server and browser, next the server starts listening to requests on a port. The browser sends requests, and the server receives them. After, the server processes the received URL and starts searching the database for it. If the server finds the requested resource, the server sends it to the browser.

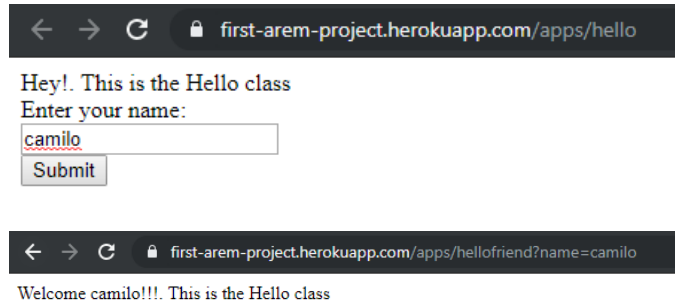
The notation package has the Web class, and it was built, so the methods that users want to execute can be obtained. It is easier to identify the methods with the Web interface. Also, the UriHandler class and the Handler interface were made with the purpose of the execution of methods.

This web application is deployed on Heroku. This is the link to the application:

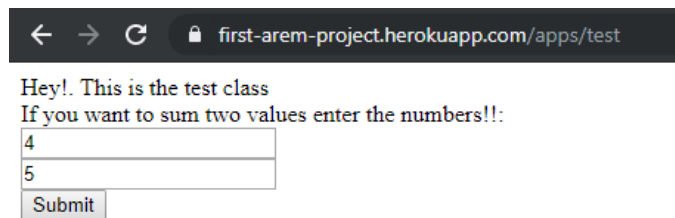
<https://first-aremproject.herokuapp.com/apps/hello>

## V. TESTS

**Testing the hello class.** This class sends a message and a form. If the button Submit is clicked on. The browser sends the entered name to a method called hellofriend, and this method returns a Welcome message. URL: <https://first-arem-project.herokuapp.com/apps/hello>

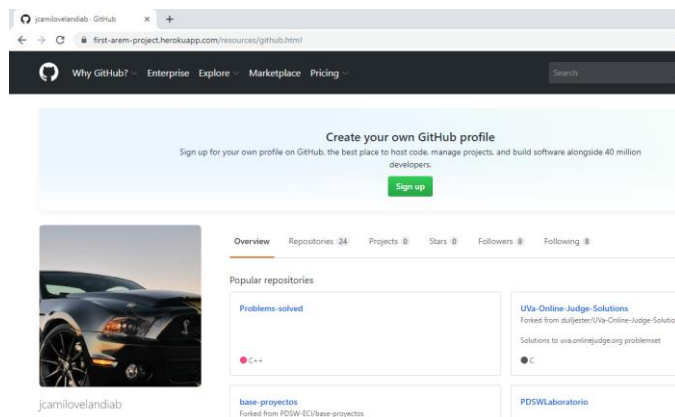


**Testing the test class.** This class sends a message and a form. This form receives two numbers. If the button Submit is clicked on. The browser sends the two entered numbers to a method called sum. This method returns the sum of two numbers. URL: <https://first-arem-project.herokuapp.com/apps/test>



**Requesting a html file.** The browser requests a html file to the server, and then the server sends the file.

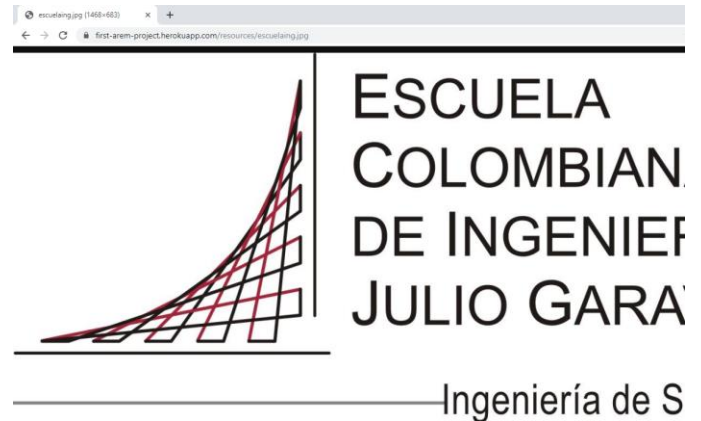
URL: <https://first-arem-project.herokuapp.com/resources/github.html>



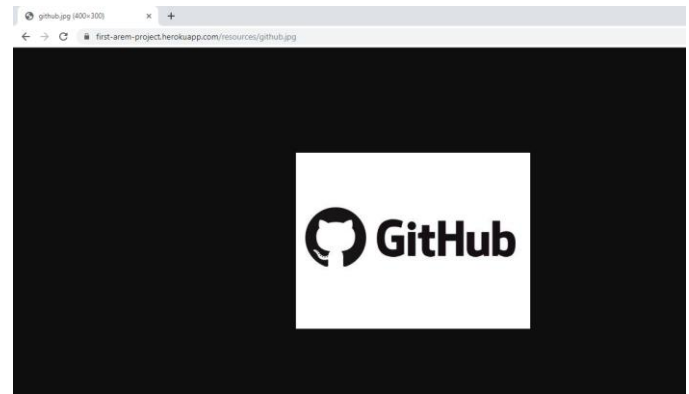
**Requesting a jpg file.**

The browser requests a jpg image to the server, and then the server sends the image.

URL: <https://first-arem-project.herokuapp.com/resources/escuelaing.jpg>



URL: <https://first-arem-project.herokuapp.com/resources/github.jpg>



## VI. CONCLUSION

In this project, the importance of the IoC framework can be noticed, and Java Plain Old Object are useful to help the construction of this sort of frameworks. As can be seen, complex constructions can be built from this framework, and many more functionalities can be added to this framework to enhance the development of our web applications.