Framework IOC y Servidor Web

Autor: *Alfonso Campuzano Amalia Inés*

# Introducción

Este documento describe una aplicación web que implementa un framework de IoC (inversión de control) para ofrecer un modelo de software Cliente-Servidor. El servidor recibe peticiones del cliente por el navegador desde la aplicación desplegada en heroku. Puede solicitar recursos estáticos: imágenes jpg y archivos html, como también permite responder a peticiones de funciones implementadas en el servidor propiamente, como generar un html a través de un método sin parámetros y con parámetros. Para el caso de parámetros, se le puede pasar un número y el servidor le responderá el cuadrado del mismo.

# ARQUITECTURA DE SOFTWARE

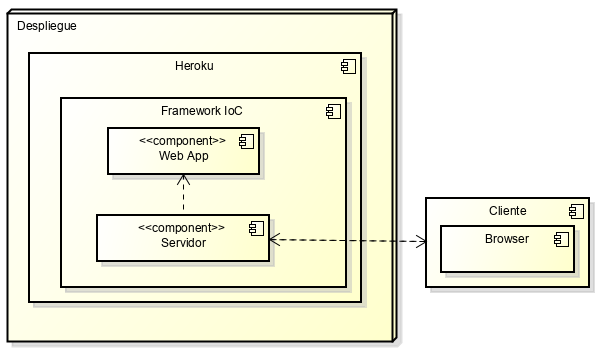
## 

Para permitir comunicación entre el cliente y el servidor, se implementó un socket para cada uno, los cuales establecen los puntos finales del enlace de comunicación y están vinculados a un puerto. El socket del servidor está vinculado al puerto de heroku (5000), y por este espera las solicitudes del cliente, ya que allí está desplegada la aplicación.

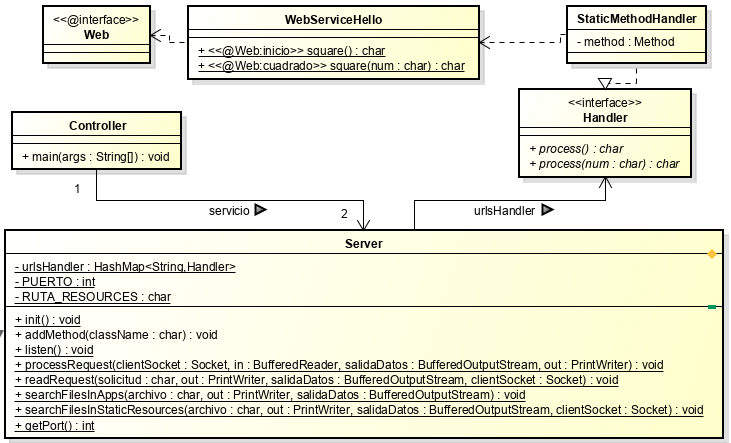
En la aplicación web se procesarán las peticiones del cliente para dar una respuesta. Dependiendo el tipo de petición, buscará en apps o en los recursos estáticos. Una vez se tiene respuesta, el servidor responde al cliente.

# ARQUITECTURA DE DESPLIEGUE

En el diagrama de despliegue se muestra que la aplicación está desplegada en heroku usando el puerto 5000. En la aplicación wub se encuentra el framework, el servidor y la aplicación que genera las respuestas para el cliente. El servidor escucha las peticiones del cliente por medio de del puerto de heroku, las procesa en la aplicación y retorna la respuesta, que puede ser un html o una imagen. Esta respuesta es dada dependiendo el requestURI, ya que se puede especificar que se busque en apps o en recursos estáticos.



# Diagrama de clases



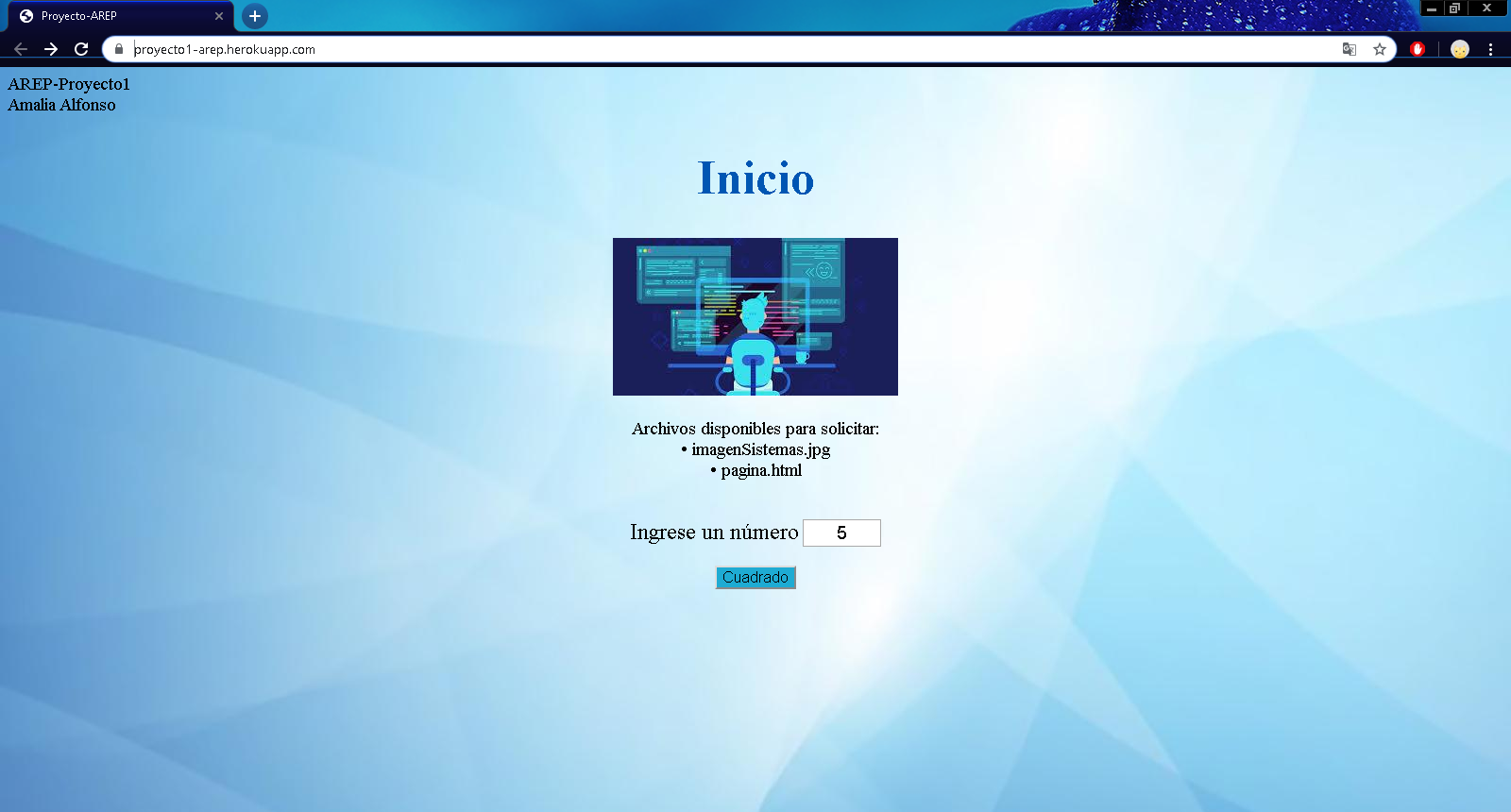
En este diagrama podemos ver la estructura del sistema, en el cual se muestran las clases con atributos y métodos. La clase en al cual se presta el servicio de http es Service la cual permite la comunicación con el cliente. En la misma, se resuelven las solicitudes del cliente.

La aplicación tiene como clase principal Controller, la cual inicializa el servidor.

# Pruebas

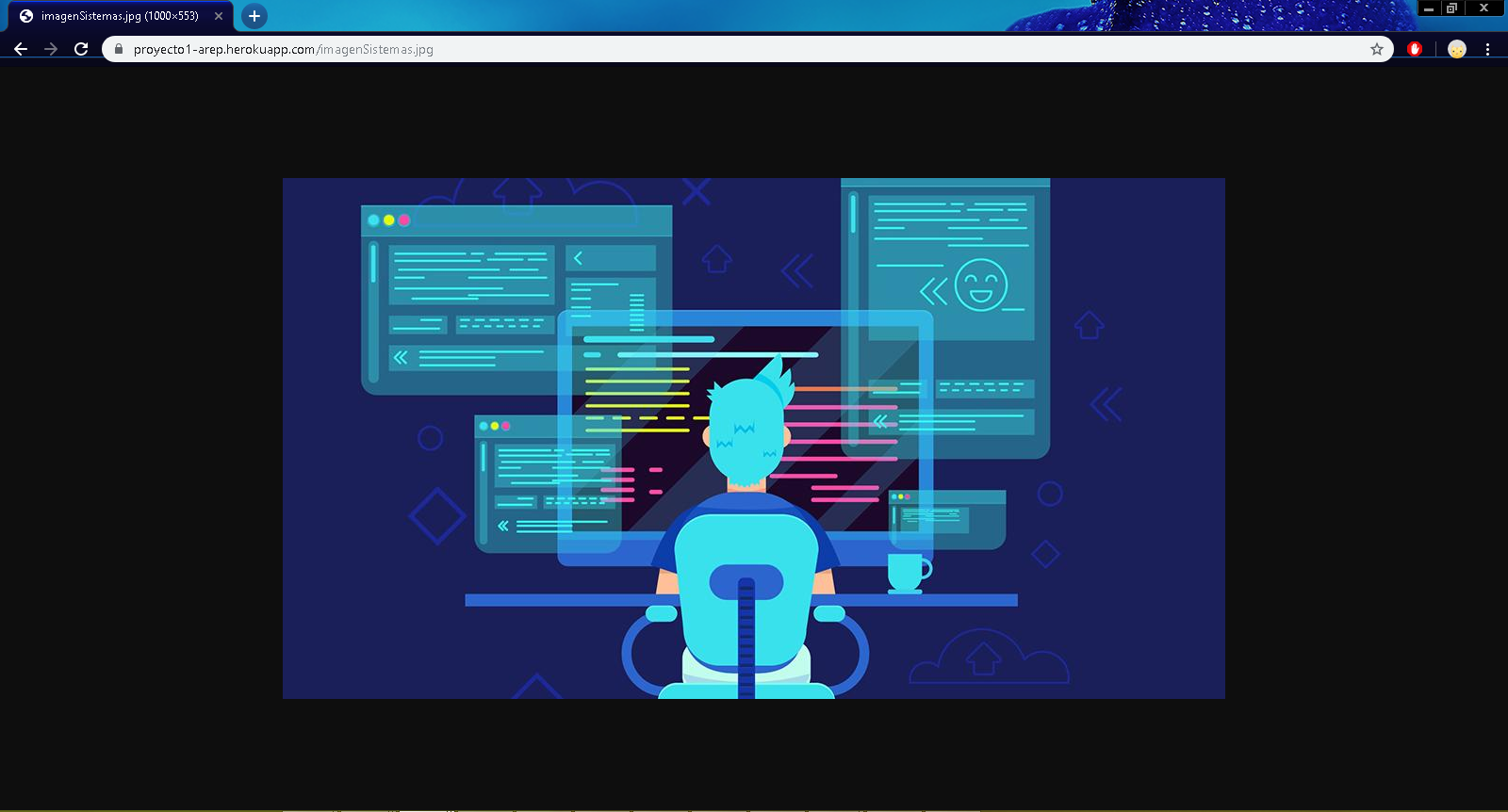
A continuación se mostrará cómo funciona la aplicación web:

Cómo ya se mencionó en el artículo, la aplicación responde a peticiones de imágenes y archivos html. Para ver su funcionalidad puede ingresar a <https://proyecto1-arep.herokuapp.com/>, en la cual se muestra la página inicial de la aplicación:



En esta página puede ver los archivos disponibles para realizar solicitudes.

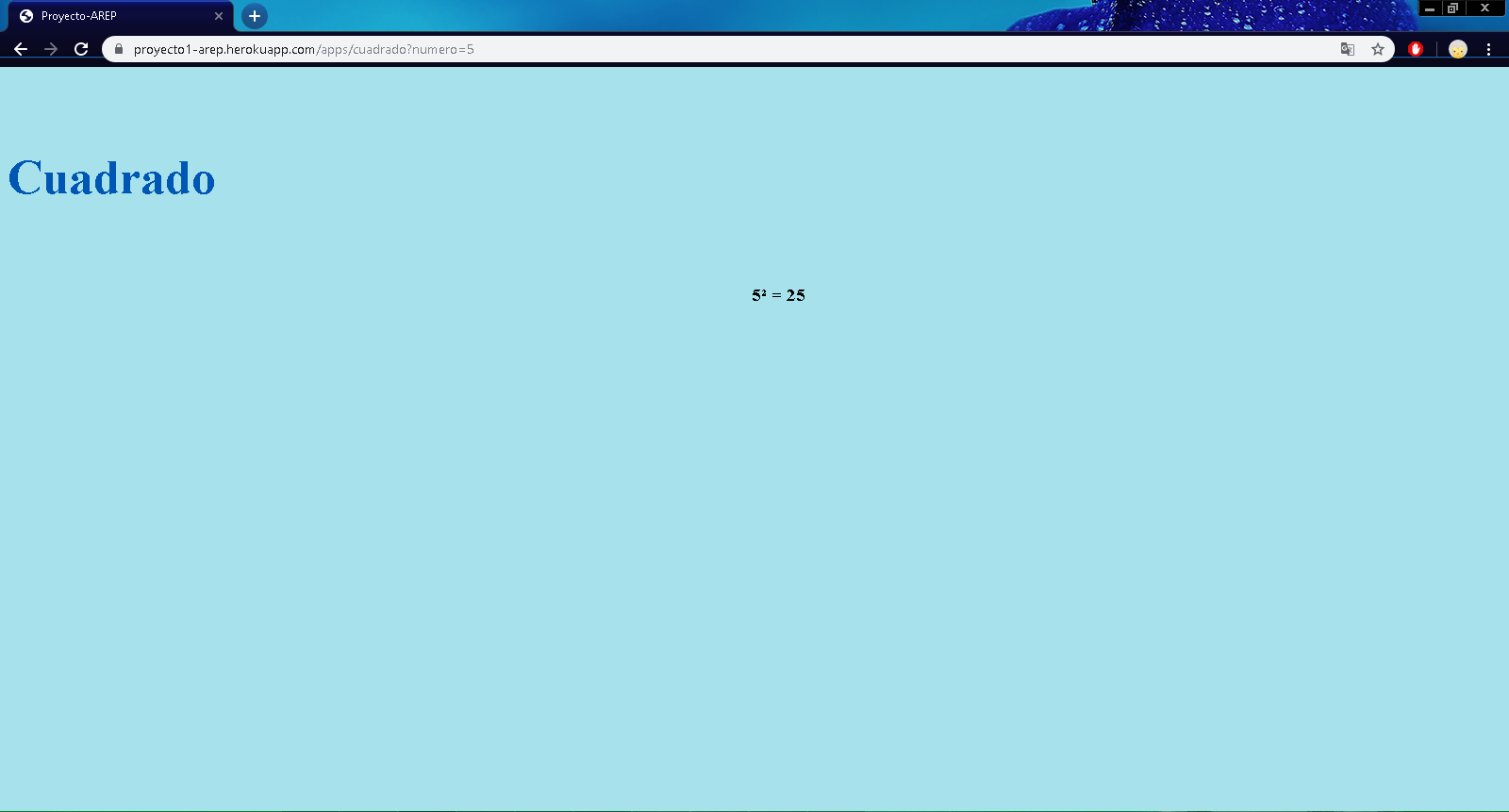
Puede solicitar un archivo o imagen disponible desde el buscador del navegador así: <https://proyecto1-arep.herokuapp.com/imagenSistemas.jpg>, o puede dar clic sobre los archivos mostrados en la página principal:



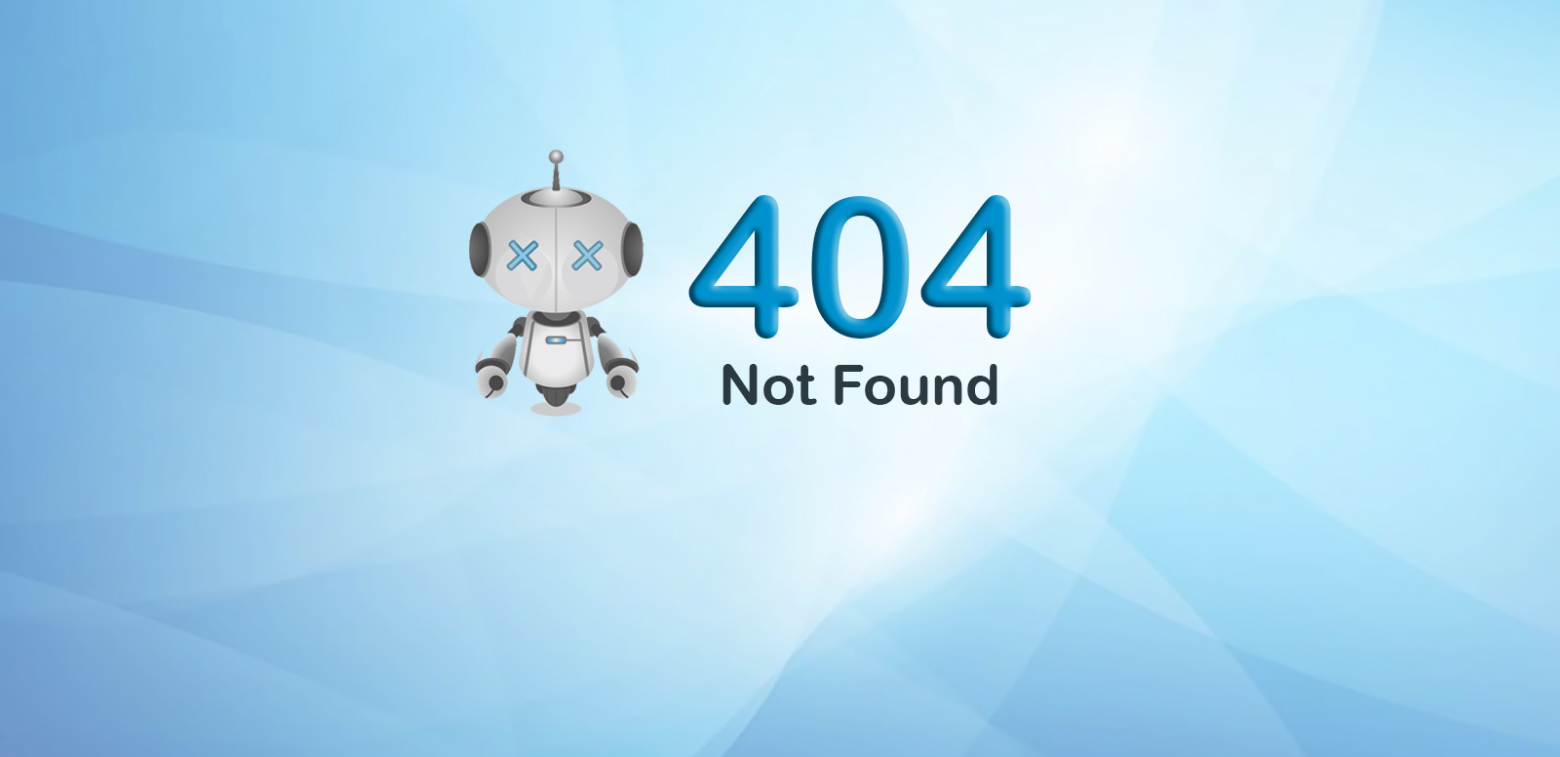


Adicionalmente, puede realizar petición de servicios que presta la aplicación. Actualmente, usted le puede ingresar un número y la aplicación le dará el resultado de elevar el número al cuadrado dirigiéndolo a otra página html. También puede solicitar ese servicio con la dirección, por ejemplo: <https://proyecto1-arep.herokuapp.com/apps/calculo?numero=5>

:



Si por error, usted solicita un archivo que no se encuentra disponible, le mostrará el siguiente mensaje:



Lo más interesante del desarrollo de esta aplicación, fue el uso de anotaciones para la realización del Framework, aquí hay una demostración de la implementación realizada:



Esta anotación permite responder a solicitudes sin parámetro, la cual es usada para mostrar la página inicial de la aplicación.



Esta anotación recibe un parámetro y responde con una cadena con el código html que se desea enviar al cliente. Si se da cuenta, es la que genera la respuesta a la solicitud realizada cuando se ingresó anteriormente un número.

# Conclusiones

En este proyecto se logró crear una aplicación web con un servidor.

Es conveniente usar pojos cuando se va a tratar con clases simples y cuando no se requiere de un framework especial.

Se implementó un framework IoC, lo cual permite gestionar los componentes libremente, y establecer las relaciones como uno desea.

# REFERENCIAS

• Carlospesquera. (2014, 19 marzo). QUÉ ES UN POJO, EJB Y UN BEAN? Recuperado de <http://carlospesquera.com/que-es-un-pojo-ejb-y-un-bean/>

• Software Guru, & Ixcoatl Pérez. (2018, 5 junio). POJOs y Frameworks Ligeros: Simplificando el Desarrollo. Recuperado de <https://sg.com.mx/content/view/234>