 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-27

SISTEMA DE SERVICIOS BASADOS EN REST PARA LA MANIPULACIÓN DE DATOS SENSIBLES

Juan Camilo Velásquez Vanegas

Ingeniería en Sistemas de Información

Andrés Felipe Giraldo Forero

INSTITUTO TECNOLÓGICO METROPOLITANO

23 DE MAYO DE 2018

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RESUMEN

En la era en que vivimos, existe una alta tendencia por la sistematización tecnológica, múltiples sectores han sido partícipes de ello, lo cual le ha permitido a las compañías mejorar día a día su competitividad en el mercado y también así mejorar los resultados obtenidos. El sector médico no ha sido ajeno a esta tendencia, tanto así que entidades de talla mundial o incluso nacional han puesto gran empeño y enfoque en estos temas, prueba de esto existen compañías nacionales con proyectos enfocados en el manejo de información clínica electrónica, que además integran el uso de estándares internacionales como por ejemplo OpenMRS y OpenEHR, herramientas que buscan proveer arquitecturas estándares para el manejo de información clínica de manera virtual.

Por otra parte, también existen proyectos de investigación locales, que buscan procesar información médica sensible por medio de herramientas de software, con el objetivo de analizar y procesar imágenes médicas y a partir de los resultados determinar patrones existentes que ayuden en la detección de ciertas enfermedades.

Sin embargo es de vital importancia saber que para realizar manipulación de cualquier información clínica, se debe tener en cuenta la privacidad y anonimato de los pacientes. Así, surge la necesidad de diseñar herramientas que cumplan con dichas características. De esta forma, y teniendo lo anterior se busca definir una arquitectura base de una herramienta tecnológica que permita brindar a los investigadores un ambiente controlado, con una serie de funcionalidades para la manipulación y consulta de información médica sensible.

El presente trabajo, se encuentra desarrollado, mediante el uso de una arquitectura orientada a servicios (SOA), más específicamente una solución implementando protocolo de servicios REST. Como principales resultados, se logra la construcción de varias funcionalidades, como por ejemplo: la creación de solicitudes de procesamiento a partir de la carga de archivos, comenzar ejecución y procesamiento en el sistema de colas del clúster a través de comandos qsub y la obtención de los resultados luego de finalizado el procesamiento realizado en el sistema de colas. Adicional a esto se derivan otros servicios que permiten la configuración previa de investigadores y la generación de reportes en formato PDF. La herramienta cuenta con una estructura de base de datos en MySQL para todo lo concerniente al manejo de investigadores, solicitudes y auditoría. Una implementación del proyecto, se encuentra disponible online en la siguiente URL <https://github.com/pillowslept/clinical-data>.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RECONOCIMIENTOS

La realización de esta investigación fue posible gracias a la oportunidad y confianza que se me brindó para materializar la idea. También al apoyo y disposición brindada por el asesor Andrés Felipe Giraldo Forero, quien siempre estuvo dispuesto a solucionar toda inquietud o necesidad que se iba presentando durante el desarrollo.

A los profesores de la carrera Juan Carlos Zuluaga y Diego Tangarife quienes fueron parte importante en mi desarrollo durante la etapa de aprendizaje de mi carrera profesional.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ACRÓNIMOS

PBS (Portable Batch System) Sistema de lote portátil, es el nombre de un software que realiza la programación de trabajos o tareas computacionales por lotes.

Spring Es un framework para el desarrollo de aplicaciones, es de código abierto y enfocado en la plataforma Java.

REST (Representational State Transfer) Transferencia de Estado Representacional, es un protocolo de intercambio y manipulación de datos en los servicios de internet.

HTTP (Hypertext Transfer Protocol) Protocolo de comunicación que permite las transferencias de información en la World Wide Web entre diferentes servicios.

Git Es un software de control de versiones, enfocado en la eficiencia y la confiabilidad del mantenimiento de versiones de código fuente.

MySQL Es un sistema de gestión de bases de datos relacional desarrollado por Oracle Corporation y considerada como la base datos de código abierto más popular del mundo.

Comando javac Es una herramienta que se encarga de leer las definiciones de clase y de interfaz, escritas en el lenguaje de programación Java, y las compila en archivos de clase bytecode.

Comando java Inicia una aplicación Java. Lo hace iniciando un entorno de tiempo de ejecución de Java, cargando una clase específica y llamando al método principal de esa clase (main).

Maven Es una herramienta de software para la gestión y construcción de proyectos Java que ayuda en el manejo de dependencias de otros módulos y componentes externos.

OpeEHR Comunidad virtual para la administración y mantenimiento de información su interoperabilidad universal.

OpenMRS Proyecto de software que se enfoca en respaldar el manejo de información clínica.

SOA (Arquitectura Orientada a Servicios) Es un estilo de arquitectura tecnológica que se apoya en la orientación a servicios.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

TABLA DE CONTENIDO

1. INTRODUCCIÓN	7
1.1 Objetivos	8
1.1.1 General	8
1.1.2 Específicos	8
2. MARCO TEÓRICO	9
3. METODOLOGÍA.....	12
4. RESULTADOS Y DISCUSIÓN.....	15
5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO	28
5.1 Conclusiones.....	28
5.2 Recomendaciones	28
5.3 Trabajo futuro	29
REFERENCIAS	30

TABLA DE FIGURAS

<i>Figura 1: Arquitectura base a implementar usando Spring Boot. (Elaboración propia)</i>	10
<i>Figura 2: Diagrama de flujos de la aplicación. (Elaboración propia)</i>	16
<i>Figura 3: Consumo servicio de creación investigadores. (Elaboración propia)</i>	18
<i>Figura 4: Respuesta de consumo servicio de creación investigadores. (Elaboración propia)</i>	19
<i>Figura 5: Consumo servicio de inactivación de investigadores. (Elaboración propia)</i>	20
<i>Figura 6: Respuesta de consumo servicio de inactivación de investigadores. (Elaboración propia)</i>	20
<i>Figura 7: Consumo de servicio de activación de investigadores. (Elaboración propia)</i>	21
<i>Figura 8: Respuesta de consumo de servicio de activación de investigadores. (Elaboración propia)</i>	21
<i>Figura 9: Contenido de archivo escrito en Python. (Elaboración propia)</i>	22
<i>Figura 10: Consumo de servicio para adjuntar archivos. (Elaboración propia)</i>	22
<i>Figura 11: Consumo de servicio para adjuntar archivos con recursos externos. (Elaboración propia)</i>	23
<i>Figura 12: Respuesta de consumo de servicio para adjuntar archivos. (Elaboración propia)</i>	23
<i>Figura 13: Consumo de servicio para comenzar procesamiento en sistema de colas. (Elaboración propia)</i>	24
<i>Figura 14: Respuesta de consumo de servicio para comenzar procesamiento en sistema de colas. (Elaboración propia)</i>	24
<i>Figura 15: Consumo de servicio para consultar el estado de una solicitud. (Elaboración propia)</i>	25
<i>Figura 16: Respuesta de consumo de servicio para consultar el estado de una solicitud en progreso. (Elaboración propia)</i>	25
<i>Figura 17: Respuesta de consumo de servicio para consultar el estado de una solicitud finalizada. (Elaboración propia)</i>	25
<i>Figura 18: Consumo de servicio para consultar la respuesta de una solicitud. (Elaboración propia)</i>	25
<i>Figura 19: Respuesta de consumo de servicio para consultar la respuesta de una solicitud. (Elaboración propia)</i>	26
<i>Figura 20: Consumo de servicio para generación de reporte por solicitud. (Elaboración propia)</i>	26
<i>Figura 21: Reporte generado para una solicitud específica. (Elaboración propia)</i>	26
<i>Figura 22: Respuesta de error al consumir un servicio. (Elaboración propia)</i>	27

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

1. INTRODUCCIÓN

Dentro del desarrollo y diseño de software, existen en el mercado infinidad de herramientas, lenguajes y frameworks los cuales buscan ofrecer diferentes alternativas para llevar a cabo soluciones tecnológicas de calidad. Por ejemplo, como lenguajes de programación importantes, podemos mencionar Python o Java, los cuales permiten el desarrollo de aplicaciones de tipo web, y que además dentro de los cuales existen una serie de herramientas y frameworks que apoyan el desarrollo, por mencionar algunos de ellos tenemos por ejemplo Flask (Grinberg, M., 2018) o Django (Holovaty, A. & Kaplan-Moss, J., 2009) para Python o como Restlet (Li, H., 2011) o Spring (Johnson, R., 2005) para Java. Básicamente todas estas herramientas y frameworks buscan facilitar en gran medida el proceso de desarrollo, agilizando los resultados, minimizando el esfuerzo y permitiendo enfocarse en lo realmente necesario, sin necesidad de tener que emplear demasiado tiempo en reinventar la rueda.

En cada capítulo de la investigación se abordarán temas que permitirán profundizar y conocer más a detalle el proceso llevado a cabo para lograr completamente los objetivos. En el capítulo 2, se sustentarán las herramientas a tener en cuenta en la investigación, además de explicar un poco sobre que son y cómo funcionan y profundizar en la arquitectura que se propone como solución y su integración con herramientas externas. En el capítulo 3, se explicará el cómo y por qué tenidos en cuenta para la elección de las herramientas usadas y se enunciarán los requisitos de infraestructura necesarios para el desarrollo. Finalmente, en el capítulo 4 se enunciarán los resultados obtenidos, se explicará paso a paso la herramienta y se mostrará por medio de un ejemplo la realización de una prueba de la aplicación.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

1.1 Objetivos

1.1.1 General

Definir una arquitectura basada en servicios REST que proporcione un ambiente controlado para el acceso y manipulación de datos clínicos.

1.1.2 Específicos

- Implementar un servicio que permita la carga de archivos al servidor con funciones diseñadas por los investigadores en lenguajes de programación Python, Java y R.
- Desarrollar una funcionalidad que permita la ejecución de los archivos a través del sistema PBS del clúster usando el comando qsub.
- Construir una serie de servicios que permitan la creación, activación o inactivación de investigadores en la herramienta.
- Diseñar un reporte que permita visualizar el resultado del procesamiento ejecutado por el clúster.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2. MARCO TEÓRICO

Mediante el uso del lenguaje de programación Java y usando frameworks de desarrollo, se pretende realizar el diseño de una herramienta tecnológica, que provea una serie de servicios usando arquitectura REST, los cuales permitan la ejecución y procesamiento de archivos, a través de un sistema de tareas por lotes y posterior a esto, obtener los resultados generados del procesamiento, para con estos generar un que agreguen valor a las tareas investigativas de quienes usen la herramienta. Estos servicios son enfocados principalmente en la manipulación y el tratamiento de información clínica sensible. En este orden de ideas, es preciso hablar sobre algunos conceptos importantes para el completo entendimiento de la investigación, para empezar el lenguaje de programación Java, el cual es un lenguaje de propósito general y cuyo enfoque es la orientación a objetos, diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible, el código que es construido en el lenguaje es compilado (usando el comando javac) y luego es interpretado por medio de un intérprete (usando el comando java) .

Dentro de los frameworks que existen actualmente para el lenguaje, existe uno muy importante llamado Spring Boot, es un proyecto creado a partir de Spring, otro framework para lenguaje de programación Java, el cual permite desarrollar y arrancar de forma muy rápida aplicaciones robustas y complejas, simplificando de esta manera el tiempo de desarrollo y la escalabilidad ofreciendo gran cantidad de utilidades y funcionalidades, enfocando el desarrollo en la realización de tres pasos fundamentales; el primero, la creación del proyecto usando gestores de dependencias como Maven o Gradle, en segundo lugar el desarrollo como tal las funcionalidades y en tercer lugar el despliegue de manera ágil en un servidor de aplicaciones web. De esta manera provee una serie de configuraciones estándar y gran cantidad de funcionalidades predefinidas, que le

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

permiten a los desarrolladores lograr resultados con altos estándares de calidad y además en poco tiempo, lo cual le apunta directamente a las necesidades del ámbito del desarrollo de software. En términos generales, el framework provee una arquitectura estándar para el diseño y la estructuración de clases y paquetes, ofreciendo de esta forma serie de notaciones propias de la herramienta, como por ejemplo, `@RestController`, `@Service`, `@Component`, `@Repository`, entre otras. Cada una de estas permite la configuración para el manejo de algún frente arquitectónico. `@Repository` provee características enfocadas en la manipulación bases de datos, tanto de tipo relacional como no relacional. `@Service`, se enfoca plenamente en orquestar todo el manejo de lógica de negocio. `@RestController` ayuda en la exposición y manipulación de los servicios como tal, es la puerta de entrada de las peticiones a las demás capas arquitectónicas.

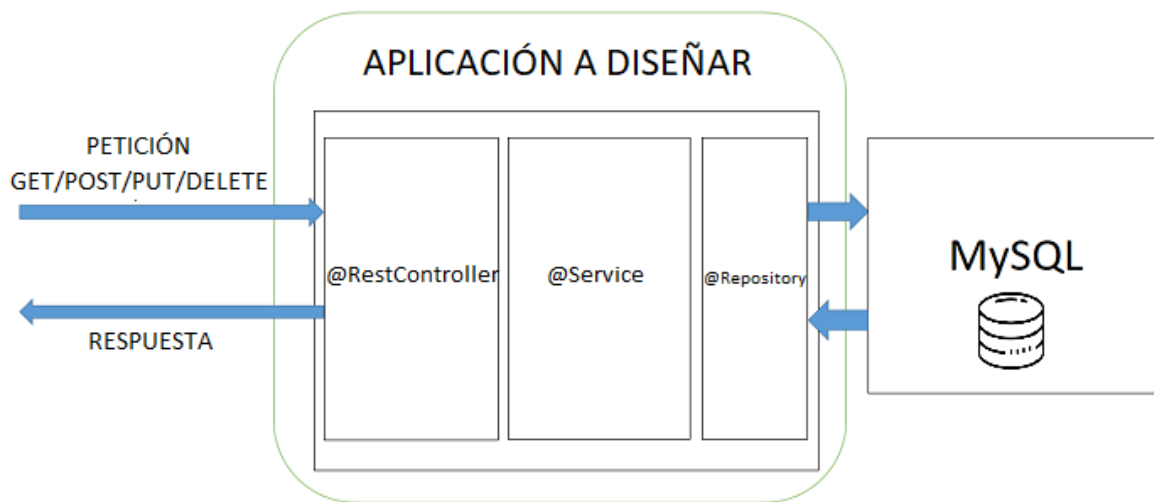


Figura 1: Arquitectura base a implementar usando Spring Boot. (Elaboración propia)

Adicional al manejo de los servicios como se menciona anteriormente, se requiere del uso de un sistema de colas externo a la aplicación, en este se busca realizar la ejecución de archivos escritos en lenguajes de programación como Java, Python o R, los cuales son enviados a través de los servicios, para esto se usa una herramienta llamada TORQUE PBS, la cual proporciona un entorno para el control sobre trabajos por lotes y recursos informáticos distribuidos. Es una herramienta de código abierto, la cual incorpora avances

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

significativos a nivel de escalabilidad, confiabilidad y funcionalidad, y actualmente se usa en decenas de miles de sitios gubernamentales, académicos y comerciales líderes en todo el mundo (Alvarruiz, F. et al, 2012).

Esta herramienta a usar es ofrecida por la institución y la documentación sobre su uso, explicación y tutoriales, pueden ser revisados en el siguiente vínculo: clusteri.itm.edu.co

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3. METODOLOGÍA

Inicialmente, como parte de la definición de las tecnologías usadas se realizó un análisis para determinar las alternativas con las que se contaba y de esta manera elegir la mejor opción. Como lenguajes de programación se planteó inicialmente realizar el desarrollo usando Python o Java, pero debido a los conocimientos más fuertes en Java, este fue el lenguaje elegido. Ahora bien, al momento de elegir la versión a usar se escogió Java 1.8, la cual provee bastantes herramientas y métodos que ayudan a facilitar el desarrollo, además de que es una de las versiones más recientes en el mercado. Ahora, más específicamente como apoyo en el lenguaje, se buscó un framework que permitiera realizar un desarrollo ágil y robusto y que además posea una buena documentación y soporte, por ello se pensó en Spring, una herramienta con la cual se cuenta con experiencia previa, más específicamente en Spring Boot, el cual es una extensión de Spring y que integra funcionalidades adicionales que le apuntan a la facilidad en la configuración y en su despliegue.

Ahora, hablando más a detalle, como parte de la arquitectura, se realiza un diseño partiendo del estándar propuesto por Spring, implementando una estructura de separación de capas y paquetes por medio del uso de notaciones propias del framework, más específicamente usando *@RestController* para la exposición de los servicios a través de una URL configurable, *@Service* para la todo lo relacionado con lógica de negocio y *@Repository* para el manejo del acceso a la base de datos por medio de una API llamada JPA, la cual provee herramientas de fácil uso para la manipulación una base de datos.

Realizar el diseño del aplicativo usando estas tecnologías, permite contar con ciertas características que le aportan un plus al resultado y que se pueden aprovechar en gran manera, una de ellas es ser una aplicación autocontenida, lo cual evita la instalación de

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

servidores web dentro del clúster de despliegue y que a futuro permita que se puedan desplegar diferentes instancias a la vez y que estas sean totalmente independientes.

Ahora bien, inicialmente se planeó la construcción de un servicio que recibiera una cadena de caracteres, la cual representaría una función diseñada por un investigador, pero que debido a la complejidad que se presentó en la manipulación de esta, se optó por buscar otras alternativas para la transferencia de esta información, y por ende se diseñó la funcionalidad de manera que en vez de recibir la cadena de caracteres, permitiera adjuntar un archivo en las extensiones *.java*, *.py* o *.r*, y de esta forma guardarlo en el disco del clúster, para realizar con este la ejecución a través de TORQUE PBS usando el comando *qsub*. Otra parte importante a mencionar, es que algunas de estas funciones que se envían a ejecutar a través de los servicios, requieren ciertos paquetes o dependencias adicionales, los cuales deben existir configurados previamente en una carpeta, es por ello que al momento de la construcción del servicio se realizaron una serie de validaciones que permitieran la ejecución de este tipo de funciones y que en caso de requerir recursos que no se encuentren previamente configurados, deberán ser solicitados y posteriormente configurados.

Otro aspecto importante a mencionar, es que la aplicación toma como punto de partida unos archivos de tipo “template”, los cuales deben existir previamente configurados por cada lenguaje soportado (Java, Python y R) y que poseen toda la configuración de comandos y requisitos que son necesarios para la ejecución de las funciones en el TORQUE PBS (Archivo con sintaxis *.sh*), algunos de sus comandos de configuración pueden ser las rutas del lenguaje a ejecutar, rutas de los archivos en los cuales escribir la respuesta obtenida, el total de nodos a usar, entre otros, esta configuración externa se realiza de esta manera debido a que para cada lenguaje se requieren configuraciones de comandos que pueden variar y que pensando en que esto pueda cambiar con el tiempo, lo ideal es que el recurso pueda ser modificado en cualquier momento sin necesidad de

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

modificar la aplicación. Este archivo deberá existir para cada lenguaje, de lo contrario no se podrá dar comienzo al proceso.

Al momento de obtener una respuesta de una función previamente enviada a ejecución, se podrá generar un PDF, para el cual se optó por usar una librería llamada ITextPDF, la cual posee amplia documentación y soporte por sus desarrolladores, además de que simplifica el diseño del mismo, porque cuenta con una serie de configuraciones de fácil entendimiento e implementación.

Finalmente, para todo el manejo de perfilamiento, configuración de investigadores, registro de las solicitudes de procesamiento y para el control y auditoría se optó por diseñar una base de datos en MySQL, en la cual se manejan la información de los investigadores y sus respectivas solicitudes de procesamiento, los estados por los cuales pueden pasar estas solicitudes e información de fechas, tanto de creación como de finalizado del procesamiento. Todo esto es de vital importancia para monitorear y realizar controles sobre el uso que se le da a la herramienta.

Ahora, como requisitos en la construcción de la herramienta, se deben cumplir los siguientes:

- Conexión a internet
- Eclipse o Spring Tool Suite (STS) como IDE de desarrollo
- Java 1.8
- Consumidor de servicios REST como Advance Rest Client o Postman
- Como variables de entorno Java, Python y R
- Acceso de lectura, escritura y ejecución de un folder definido

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

4. RESULTADOS Y DISCUSIÓN

Como resultado, se obtiene una aplicación web que puede ser compilada y empaquetada en formato WAR o JAR y que simplemente deberá ser desplegada en un servidor que corra aplicaciones Java. Adicionalmente, la aplicación cuenta con un paquete de archivos enfocados en realizar pruebas automatizadas del comportamiento y funcionamiento del código bajo ciertas circunstancias definidas, los cuales son llamados pruebas unitarias.

Para la realización de un flujo en la aplicación, se deben tener en cuenta las siguientes condiciones para lograr con éxito la ejecución:

- El investigador deberá crearse como usuario de la herramienta.
- Se debe contar con un archivo en formato .java, .py o .r, el cual debe poseer una estructuración válida acorde al lenguaje de programación (Su sintaxis, sus condiciones y demás).
- En caso de requerir recursos adicionales (como jars, paquetes o librerías externas) deberán estar configurados dentro de las carpetas definidas.
- Debe existir configurado el “template” con las condiciones de ejecución al clúster para el lenguaje a usar.

Igualmente, para conocer un poco el flujo de la aplicación podemos referirnos al siguiente diagrama, el cual muestra el funcionamiento en términos generales y su comportamiento en cada paso del flujo:

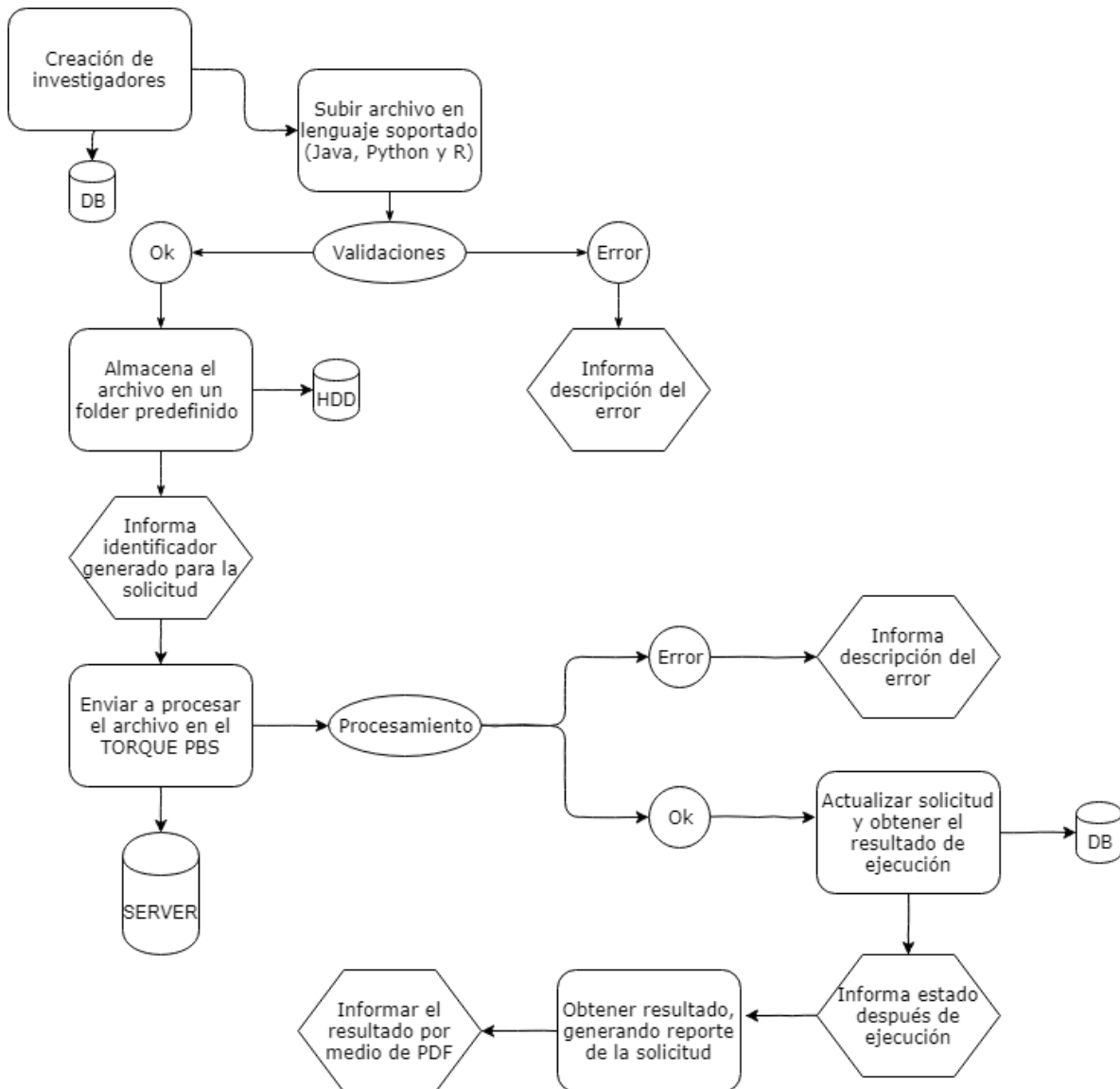


Figura 2: Diagrama de flujos de la aplicación. (Elaboración propia)

Para comenzar, los servicios expuestos por la aplicación deberán ser consumidos de la siguiente manera:

<http://host:puerto/>

- host: Nombre del servidor donde se encuentre desplegada la aplicación.
- puerto: Número del puerto configurado en la aplicación por el cual se desplegarán los servicios, por defecto el puerto 9000.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Por ejemplo, supongamos que el nombre del host es *clusteritm* y que el puerto de despliegue de la aplicación es el *9000*, la url base de consumo quedaría de la siguiente manera: <http://clusteritm:9000/>

(Para efectos de realización de pruebas, se usaría de la siguiente forma: <http://localhost:9000/ClinicalData/api/>)

Ahora, para complementar la URL debemos saber que el path siguiente a la definición del host y su puerto es la que define la URL de consumo, por ejemplo continuando con el ejemplo anterior y usando como nombre *ClinicalData/api/*, quedaría de la siguiente forma: <http://clusteritm:9000/ClinicalData/api/>

Continuando después del path anterior, se derivan otros subpaths, los cuales hacen referencia a cada funcionalidad que puede ser consumida, a través de su respectivo método de consumo (POST, PUT, DELETE, GET), así:

- */investigator/create/ PUT*
- */investigator/activate/{identifier} POST*
- */investigator/inactivate/ POST*
- */file/upload POST*
- */processData/start POST*
- */processData/state/{identifier} GET*
- */processData/result/{identifier} GET*
- */report/request/{identifier} GET*
- */report/investigator/{investigatorId} GET*

Todos estos servicios, reciben como entrada, una serie de parámetros que deben estar estructurados en formato JSON. Además, tienen como retorno un mensaje explicativo

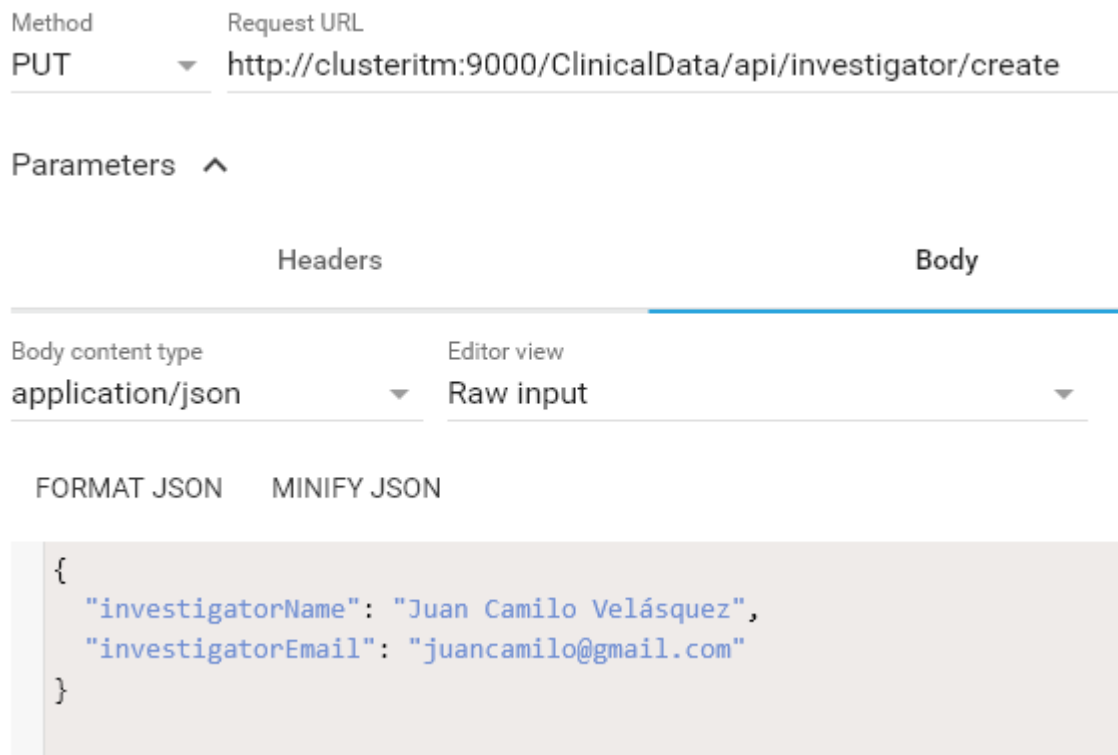
	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

junto con un código, el cual identifica si la petición fue exitosa o si falló por alguna razón; dentro de posibles códigos se encuentran:

- 200 Procesada con éxito.
- 400 No procesada por validación de negocio.
- 500 Error interno del servidor.

Ahora, usando Advance Rest Client o una aplicación similar (Como Postman, Insomnia REST Client, entre otros), se puede realizar el consumo de todos los servicios expuestos.

Para la creación de investigadores se consumirá el servicio `/investigator/create/`, de la siguiente forma:



The screenshot shows an API client interface with the following details:

- Method:** PUT
- Request URL:** `http://clusteritm:9000/ClinicalData/api/investigator/create`
- Parameters:** Expanded to show:
 - Body content type:** application/json
 - Editor view:** Raw input
- Body:** A JSON object:


```
{
  "investigatorName": "Juan Camilo Velásquez",
  "investigatorEmail": "juancamilo@gmail.com"
}
```
- Buttons:** FORMAT JSON and MINIFY JSON

Figura 3: Consumo servicio de creación investigadores. (Elaboración propia)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Como respuesta de este servicio, se obtendrá una información sobre el investigador creado, como lo es el estado actual y el ID, el cual lo identificará dentro de la aplicación, de la siguiente manera:



```

200 OK 572.50 ms
{
  "id": 1,
  "name": "Juan Camilo Velásquez",
  "email": "juancamilo@gmail.com",
  "state": "ACTIVO"
}

```

Figura 4: Respuesta de consumo servicio de creación investigadores. (Elaboración propia)

Adicional al servicio de creación de investigadores, existen otros dos servicios enfocados en el manejo de los usuarios, los cuales se encargan de activar o inactivar las cuentas, de esta manera se podría restringir o permitir el acceso a los demás servicios en el momento que sea requerido.

Para realizar la inactivación de investigadores se consume el servicio `/investigador/inactivate/`, de la siguiente manera:

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22





Method POST Request URL <http://clusteritm:9000/ClinicalData/api/investigator/inactivate>

Parameters ^

Headers	Body
Body content type application/json	Editor view Raw input
FORMAT JSON MINIFY JSON	<pre>{ "investigatorId": 1 }</pre>

Figura 5: Consumo servicio de inactivación de investigadores. (Elaboración propia)

200 OK 75.00 ms

El investigador con identificador <1> fue inactivado con éxito

Figura 6: Respuesta de consumo servicio de inactivación de investigadores. (Elaboración propia)

Para realizar la activación de investigadores se consume el servicio `/investigator/activate/`, de la siguiente manera:

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Method Request URL
 POST ▼ http://clusteritm:9000/ClinicalData/api/investigator/activate





Parameters ^

Headers	Body
Body content type application/json ▼	Editor view Raw input ▼
FORMAT JSON MINIFY JSON	
<pre>{ "investigatorId": 1 }</pre>	

Figura 7: Consumo de servicio de activación de investigadores. (Elaboración propia)

200 OK

10.70 ms

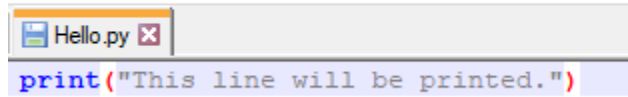





El investigador con identificador <1> fue activado con éxito

Figura 8: Respuesta de consumo de servicio de activación de investigadores. (Elaboración propia)

Ahora, existiendo el investigador, se puede realizar la petición al servidor de crear una solicitud de procesamiento y adjuntar un archivo que se desea ejecutar, por ejemplo para consumir el servicio /file/upload se usará un archivo *Hello.py* (Escrito en Python) y se usará el id del investigador previamente creado, con esto se procedería a consumir el servicio de la siguiente forma:

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



```

Hello.py
print("This line will be printed.")

```

Figura 9: Contenido de archivo escrito en Python. (Elaboración propia)

Method: POST Request URL: http://localhost:9000/ClinicalData/api/file/upload

Parameters ^


Headers	Body	Variables
Body content type: multipart/form-data		
		
Field name: file	<input type="button" value="CHOOSE FILE"/> Hello.py (35 bytes)	
Content type (Optional)		
investigatorId	1	

Figura 10: Consumo de servicio para adjuntar archivos. (Elaboración propia)

En caso de que el archivo a procesar, requiera de librerías externas adicionales, se deberán enviar un parámetro adicional en la petición, en el cual se definen los recursos requeridos, todos se envían separados por comas (,) y el servicio validará que estos recursos si existen configurados dentro del servidor para el lenguaje específico, así:

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Method: POST Request URL: http://localhost:9000/ClinicalData/api/file/upload

Parameters ^

Body content type: multipart/form-data

Field name: file Hello.py (35 bytes)

Content type (Optional):

investigatorId: 1

Content type (Optional):

resources: recurso1,recurso2,recurso3

Figura 11: Consumo de servicio para adjuntar archivos con recursos externos. (Elaboración propia)

Este servicio genera como respuesta un identificador único de 36 caracteres, como se muestra a continuación:

200 OK 259.30 ms

El archivo ha sido almacenado con éxito, identificador generado para la solicitud: <c42bfa6c-8f70-4af7-8fbc-2969d2b3b3a2>.

Figura 12: Respuesta de consumo de servicio para adjuntar archivos. (Elaboración propia)

Con este id que genera como respuesta este servicio, se va a identificar la solicitud en los demás servicios o procesos, además el sistema creará todo lo necesario para enviar al sistema TORQUE PBS el archivo solicitado.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Luego de haber ejecutado este servicio, usando el id del investigador y el identificador generado para la solicitud, se procede a ejecutar el servicio /processData/start, este servicio es el que se encarga de comenzar el procesamiento en el sistema de colas del archivo previamente cargado, este servicio se consumiría de la siguiente manera:

Method Request URL

POST ▼ http://localhost:9000/ClinicalData/api/processData/start





Parameters ^

Headers	Body
Body content type application/json ▼	Editor view Raw input ▼
FORMAT JSON MINIFY JSON	
<pre> { "identifier": "c42bfa6c-8f70-4af7-8fbc-2969d2b3b3a2", "investigatorId": 1 } </pre>	

Figura 13: Consumo de servicio para comenzar procesamiento en sistema de colas. (Elaboración propia)

Al ejecutar el servicio, el sistema informará que la solicitud ha comenzado a ser ejecutada por el sistema de colas, de la siguiente forma:

200 OK 352.80 ms DETAILS ▼

Investigador <Juan Camilo Velásquez>, la solicitud <c42bfa6c-8f70-4af7-8fbc-2969d2b3b3a2> ha comenzado a ser procesada por el cluster.

Figura 14: Respuesta de consumo de servicio para comenzar procesamiento en sistema de colas. (Elaboración propia)

En este momento, la aplicación ha realizado toda la configuración necesaria para que el archivo que enviamos a ejecutar sea procesado por medio del sistema de colas, además,

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

mientras el sistema de colas se encuentra ejecutando las funciones enviadas, se puede monitorear el estado en el que se encuentra la solicitud, por medio del servicio `/processData/state/{identifiier}`, de la siguiente forma:

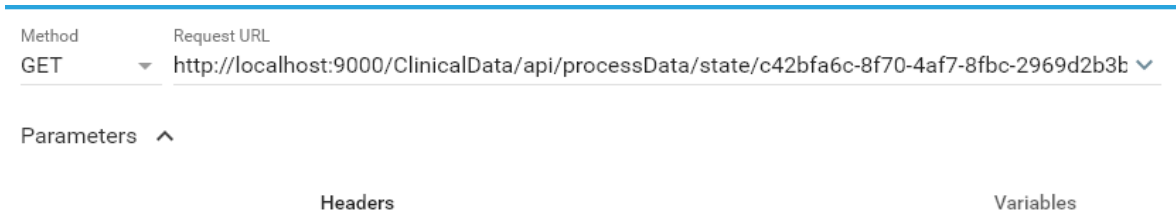


Figura 15: Consumo de servicio para consultar el estado de una solicitud. (Elaboración propia)

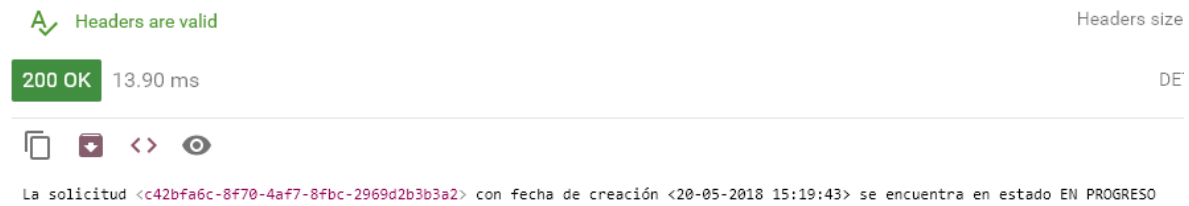


Figura 16: Respuesta de consumo de servicio para consultar el estado de una solicitud en progreso. (Elaboración propia)

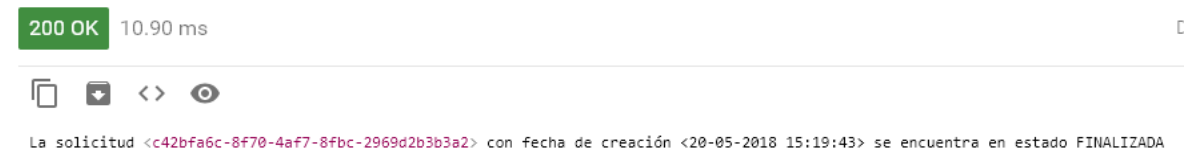


Figura 17: Respuesta de consumo de servicio para consultar el estado de una solicitud finalizada. (Elaboración propia)

Cuando se consume el servicio anterior, y su respuesta es que se encuentra en esta Finalizado, se podría ejecutar otro servicio que se encarga de consultar la respuesta que arrojó la solicitud de procesamiento, para ello se consume el servicio `/processData/result/{identifiier}` de la siguiente forma:

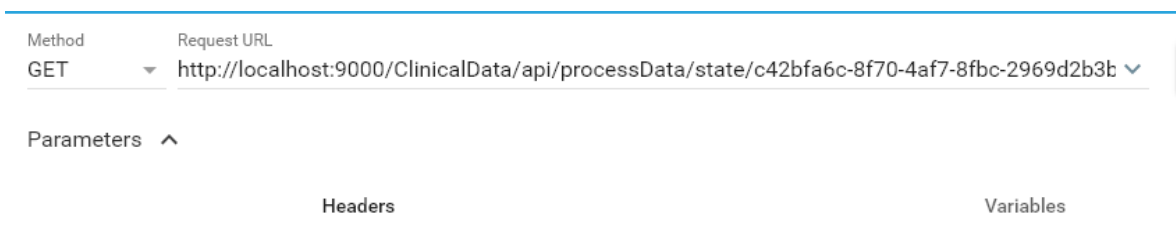


Figura 18: Consumo de servicio para consultar la respuesta de una solicitud. (Elaboración propia)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

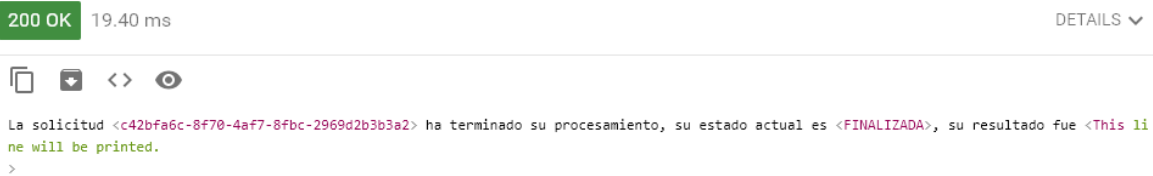


Figura 19: Respuesta de consumo de servicio para consultar la respuesta de una solicitud. (Elaboración propia)

Como se puede ver en la imagen anterior, este servicio muestra el resultado luego de ejecución del servicio en el sistema de colas, este resultado se puede complementar consumiendo un servicio que permite visualizar la solicitud con su detalle de una mejor forma luego de terminada su ejecución, este servicio genera un reporte en PDF y para mejor visualización, se puede consumir a través de un explorador web como Chrome, Firefox, entre otros, el servicio es /report/request/{identifiier}, se consume así:

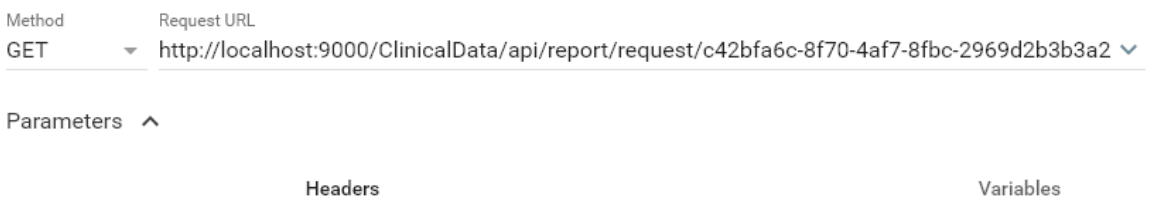


Figura 20: Consumo de servicio para generación de reporte por solicitud. (Elaboración propia)

	REPORTE DE PROCESAMIENTO DE SOLICITUD		F. de generación: 20-05-2018 15:49:51	
			Investigador: Juan Camilo Velásquez	
			Email: juancamilo@gmail.com	
Id	Identificador	F. creación	Archivo	Estado
1	c42bfa6c-8f70-4af7-8fbc-2969d2b3b3a2	20-05-2018 15:19:43	Hello.py	FINALIZADA
Resultado: This line will be printed.				

Figura 21: Reporte generado para una solicitud específica. (Elaboración propia)

Luego de haber ejecutado esta serie de servicios, se habrá concluido un flujo normal dentro de la aplicación, cada servicio de los mencionados posee una serie de validaciones que deben pasar antes de ejecutar la funcionalidad, a continuación se muestra una

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

imagen de ejemplo, sobre cómo serán los mensajes mostrados y su respectivo código en caso de que se envíe información no válida a través de la aplicación:

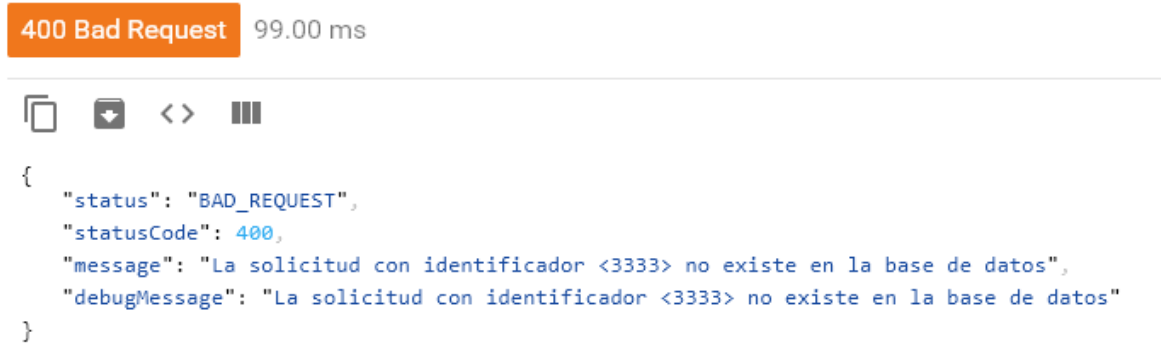


Figura 22: Respuesta de error al consumir un servicio. (Elaboración propia)

Finalmente, todo el insumo obtenido como resultado de la investigación, se encuentra alojado en un repositorio público versionado mediante la herramienta Github, allí pueden encontrarse todos los recursos desarrollados durante la investigación:

<https://github.com/pillowslept/clinical-data>

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

5.1 Conclusiones

En el presente trabajo se diseñó una aplicación web, cuya principal característica es la manipulación, procesamiento y consulta de información clínica sensible, sin necesidad de comprometer el anonimato de los pacientes.

- Se construyó un servicio que permite la carga de archivos en formatos como Java, Python y R, los cuales son enviados a procesar a través de un sistema de manejo de colas y tareas externo.
- Se elaboró una funcionalidad que permite ejecutar archivos previamente escritos en lenguajes de programación, por medio de una herramienta llamada TORQUE PBS y de esta manera obtener el resultado de su ejecución.
- Se implementaron una serie de servicios que permiten la configuración de investigadores, para permitirles el acceso a las demás funcionalidades de la aplicación y de esta manera llevar un control de todo lo que es procesado.
- Se diseñaron una serie de reportes en formato PDF, para la visualización de un resultado de procesamiento en el sistema de TORQUE PBS, además de permitir visualizar información básica del procesamiento.

5.2 Recomendaciones

- Realizar la configuración de puertos y de permisos necesarios, que permitan que la aplicación pueda estar expuesta a través de una URL, y que de esta manera sea posible realizar el consumo de los servicios desde afuera del servidor, incluso desde internet.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

5.3 Trabajo futuro

- Diseñar una interfaz web que consuma los servicios desarrollados, con el fin de que los investigadores puedan usarla para realizar los procesamientos y solicitudes a través de ella.
- Mejorar la implementación que existe actualmente para mostrar la respuesta de un procesamiento, además de incluir gráficas o alguna otra herramienta visual que soporten mejor el entendimiento del resultado obtenido y que estén enfocadas en brindar mejores herramientas al investigador para realizar su labor.
- Implementar un sistema de seguridad para la aplicación, de manera que exista un punto de acceso y un punto de salida a la misma.
- Implementar la funcionalidad de soportar la ejecución de archivos que requieren recursos o librerías externas de una versión específica, debido a que actualmente el campo que indica la versión es recibido por la aplicación pero este aún no tiene su lógica desarrollada.
- Diseñar otros reportes en PDF que ayuden a generar informes y estadísticas de los tiempos de respuesta en las ejecuciones enviadas al sistema de colas.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

REFERENCIAS

Alvarruiz, F., de Alfonso, C., Caballer, M., & Hernández, V. (2012, July). An energy manager for high performance computer clusters. In *Parallel and Distributed Processing with Applications (ISPA), 2012 IEEE 10th International Symposium on* (pp. 231-238). IEEE.

Li, H. (2011, September). RESTful Web service frameworks in Java. In *Signal processing, communications and computing (ICSPCC), 2011 IEEE international conference on* (pp. 1-4). IEEE.

Johnson, R. (2005). J2EE development frameworks. *Computer*, 38(1), 107-110.

Holovaty, A., & Kaplan-Moss, J. (2009). *The definitive guide to Django: Web development done right*. Apress.

Grinberg, M. (2018). *Flask web development: developing web applications with python*. " O'Reilly Media, Inc."

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

FIRMA ESTUDIANTES Juan Camilo Velásquez

FIRMA ASESOR Andrés Felipe Caudelo Jarama

FECHA ENTREGA: 23-05-2018

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD _____

RECHAZADO___ ACEPTADO___ ACEPTADO CON MODIFICACIONES___

ACTA NO. _____
 FECHA ENTREGA: _____

FIRMA CONSEJO DE FACULTAD _____

ACTA NO. _____
 FECHA ENTREGA: _____

ITM Institución Universitaria		MODALIDAD TRABAJO DE GRADO PRODUCTO OBTENIDO EN TALLERES O LABORATORIOS DEL ITM			Código	FDE 146				
		Registro de actividades y cumplimiento de horas / Talleres o Laboratorios de DOCENCIA			Versión	02				
					Fecha	2015-09-30				
Documento de Identidad:		1128456620								
Nombre completo del estudiante:		Juan Camilo Velásquez Vanegas								
Programa académico ITM:		Ingeniería en Sistemas de Información								
Nombre completo del Docente Asesor:		Andrés Felipe Giraldo Forero								
Fecha de iniciación del producto (aaaa/mm/dd):		01/03/2018	Fecha de terminación del producto (aaaa/mm/dd):		23/05/2018					
Nombre Taller o Laboratorio:		Sistema de servicios basados en REST para la manipulación de datos sensibles								
Ubicación:		Medellín, Antioquia								
Campus:		ITM Fraternidad								
Fecha		Actividad desempeñada por el estudiante			Hora Ingreso	Hora salida	Total horas	Firma Laboratorista	Firma Estudiante	
A	M	D								
18	3	1	Creación de repositorio y definición de herramientas a usar			4	6	2	<i>Andrés Felipe Giraldo Forero</i>	<i>Juan Camilo Velásquez</i>
18	3	6	Inicialización del proyecto, configuración de ambiente.			2	4	2	<i>Andrés Felipe Giraldo Forero</i>	<i>Juan Camilo Velásquez</i>
18	3	8	Configuración de dependencias y librerías (Junit, Mockito, Spring, JPA, entre otros)			4	6	2	<i>Andrés Felipe Giraldo Forero</i>	<i>Juan Camilo Velásquez</i>
18	3	13	Definición de la estructura a desarrollar			2	4	2	<i>Andrés Felipe Giraldo Forero</i>	<i>Juan Camilo Velásquez</i>
18	3	15	Definición de objetos y entidades como Investigador, Solicitud, Estados, entre otros.			4	6	2	<i>Andrés Felipe Giraldo Forero</i>	<i>Juan Camilo Velásquez</i>
18	3	20	Diseño de métodos de validación y personalización de secciones			2	4	2	<i>Andrés Felipe Giraldo Forero</i>	<i>Juan Camilo Velásquez</i>
18	3	22	Delimitación de lenguajes soportados			4	6	2	<i>Andrés Felipe Giraldo Forero</i>	<i>Juan Camilo Velásquez</i>
18	3	27	Control y manejo de excepciones en la aplicación			2	4	2	<i>Andrés Felipe Giraldo Forero</i>	<i>Juan Camilo Velásquez</i>
18	3	29	Creación de clases y métodos de tipo Logger y Mapper			4	6	2	<i>Andrés Felipe Giraldo Forero</i>	<i>Juan Camilo Velásquez</i>
18	4	3	Creación de métodos para validaciones de identificadores			2	4	2	<i>Andrés Felipe Giraldo Forero</i>	<i>Juan Camilo Velásquez</i>

23-05-2018
Andrés Felipe Giraldo Forero

18	4	5	Métodos y validación de lenguajes soportados como Java, Python y R	4	6	2	<i>Juan Camilo Velásquez</i>
18	4	10	Archivos para manejo de creación de carpetas y procesos para guardar en disco duro	2	4	2	<i>Juan Camilo Velásquez</i>
18	4	12	Creación de capa de acceso a datos para solicitudes de procesamiento	4	6	2	<i>Juan Camilo Velásquez</i>
18	4	17	Enumeración de lenguajes y método para creación de archivos con nombres únicos	2	4	2	<i>Juan Camilo Velásquez</i>
18	4	18	Se modifica bloque de procesamiento y se valida previa existencia de solicitud a procesar	4	6	2	<i>Juan Camilo Velásquez</i>
18	4	20	Se definen y crean clases para la ejecución de comandos usando consola del sistema	2	4	2	<i>Juan Camilo Velásquez</i>
18	4	24	Creación de servicio REST para acciones sobre investigadores (Crear, activar e inactivar)	4	6	2	<i>Juan Camilo Velásquez</i>
18	4	26	Se crean test unitarios para flujos relacionados con los investigadores	2	4	2	<i>Juan Camilo Velásquez</i>
18	4	27	Exposición de servicios para permitir carga de archivos planos en Java, Python y R	4	6	2	<i>Juan Camilo Velásquez</i>
18	5	2	Ajuste de servicios para implementar nueva estructuración al subir archivos	2	4	2	<i>Juan Camilo Velásquez</i>
18	5	4	Modificar entidades para guardar información sobre fechas de creación y actualización	4	6	2	<i>Juan Camilo Velásquez</i>
18	5	8	Modificación de nombres de atributos de parámetros de entrada	2	4	2	<i>Juan Camilo Velásquez</i>
18	5	9	Creación de métodos para consulta y manipulación de investigadores	4	6	2	<i>Juan Camilo Velásquez</i>
18	5	10	Mejora en mensajes de retroalimentación en varios servicios	2	4	2	<i>Juan Camilo Velásquez</i>
18	5	11	Modificar entidades para guardar información sobre fechas de creación y actualización	4	6	2	<i>Juan Camilo Velásquez</i>
18	5	15	Se adicionan parámetros nuevos al objeto de subir archivos	2	4	2	<i>Juan Camilo Velásquez</i>
18	5	17	Se adiciona campo email en los datos de investigador	4	6	2	<i>Juan Camilo Velásquez</i>

8106-50-52

18	5	18	Se limita el tamaño de archivos a cargar	2	4	2	<i>Juan Camilo Velásquez</i>	<i>Juan Camilo Velásquez</i>
18	5	21	Se implementa funcionalidad para el soporte de métodos de tipo Async	6	8	2	<i>Juan Camilo Velásquez</i>	<i>Juan Camilo Velásquez</i>
18	5	21	Creación de DTO para representar respuesta de ejecución por consola	2	4	2	<i>Juan Camilo Velásquez</i>	<i>Juan Camilo Velásquez</i>
18	5	22	Modificación de entidad para manejar respuesta de ejecución de comandos	2	4	2	<i>Juan Camilo Velásquez</i>	<i>Juan Camilo Velásquez</i>
18	5	22	Se crea nuevo servicios para llamado de métodos del cluster de forma asincrónica	4	6	2	<i>Juan Camilo Velásquez</i>	<i>Juan Camilo Velásquez</i>
18	5	23	Se expone servicio de reportes en PDF para investigador	2	4	2	<i>Juan Camilo Velásquez</i>	<i>Juan Camilo Velásquez</i>
TOTAL HORAS								64

23-05-2018
 J. Velásquez

Juan Camilo Velásquez
 Firma Estudiante

Juan Camilo Velásquez
 Nombre y firma Laboralista

Nombre y firma Profesional Universitario - Centro de Laboratorios

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

 Institución Universitaria	FORMATO PARA LA FORMALIZACIÓN DEL PRODUCTO OBTENIDO EN TALLERES O LABORATORIOS DEL ITM	Código	FDE 026
		Versión	01
		Fecha	2015-09-30

Fecha: 1 de marzo del 2018

Nombres y apellidos: Juan Camilo Velasquez Vanegas
 Cédula: 1128456620 Carné : 15119034
 Teléfonos: 3113917557
 Programa: Ingeniería de sistemas
 Fecha de iniciación del producto (aaaa/mm/dd): 2018/03/01
 Fecha de terminación del producto (aaaa/mm/dd): 2018/07/01
 Docencia: o Investigación
 Nombre del Taller o Laboratorio: Sistema de servicios basados en REST para la manipulación de datos sensibles.

Campus: Fraternidad

Nombre del docente asesor: Andrés Felipe Giraldo Forero Cargo: Docente Ocasional

E - Mail: feligiraldog@itm.edu.co

Diligencie el siguiente campo:

A. Descripción del producto a desarrollar:

Resumen ejecutivo: (es un breve análisis de los aspectos más importantes del Trabajo de Grado, el cual describe el producto y sus beneficiarios, el contexto, los resultados esperados, las necesidades de financiamiento y las conclusiones generales).

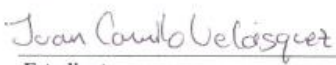
Dado las políticas de tratamiento de datos médicos, se requiere desarrollar nuevos sistemas de manipulación y consulta. Con el fin de brindarle a los investigadores herramientas de acceso y manipulación de datos sin comprometer el anonimato de los pacientes. El presente producto de laboratorio está enfocado en la definición de una arquitectura basada en servicios REST que proporcione un ambiente controlado para el acceso y manipulación de datos clínicos.

B. Detalle claramente las evidencias o anexos a entregar al finalizar el Trabajo de Grado:

La evidencia será el desarrollo de un sistema de servicios basados en REST que permita a investigadores manipular datos sin necesidad de tener acceso a los mismos. Los códigos fuentes de este desarrollo se alojarán en un repositorio Github administrado por el laboratorio de máquinas inteligentes y reconocimiento de patrones.

Nota: Entregar a los ocho (8) días de su aprobación, en el Departamento Académico al cual se encuentra adscrito.

Firmas:


Juan Camilo Velasquez
 Estudiante


Andrés Felipe Giraldo Forero
 Docente Asesor

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

 Institución Universitaria	EVALUACIÓN DE MODALIDAD TRABAJO DE GRADO Y PRÁCTICAS PROFESIONALES	Código	FDE 090
		Versión	04
		Fecha	2015-10-05

INFORMACIÓN DEL TRABAJO DE GRADO

1. Título:

Sistema de servicios basados en REST para la manipulación de datos sensibles			
Programa Académico:	Tecnología	Ingeniería	X

2. Modalidad Trabajo de Grado:

Proyecto de Grado	Práctica Profesional	Emprendimiento	
Producto de Investigación	Producto obtenido en Talleres o Laboratorios ITM	X	Pasantías
Certificación	Reconocimiento Laboral	Cursos de Posgrado	
Ingeniería para la Gente			
Grupo de investigación: Automática, Electrónica y Ciencias Computacionales		Código proyecto	
Tipo de Informe	Propuesta de Proyecto de Grado	Informe Final de Proyecto de Grado	X

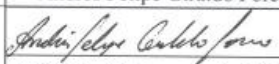
3. Información estudiante(s):

Nombre	Cédula	Correo electrónico
Juan Camilo Velásquez Vanegas	1128456620	juanvelasquez165511@correo.itm.edu.co

4. Información asesor:

Nombre	Institución	Correo electrónico
Andrés Felipe Giraldo Forero	ITM	felipegiraldo@itm.edu.co

CONCEPTO DEL JURADO EVALUADOR

Concepto inicial sobre el trabajo de grado			
Aprobado sin modificaciones	X	Se requieren modificaciones	Mención honorífica
Observaciones			
Se puede anexar hojas adicionales para una descripción más amplia de las observaciones. Justificar en esta parte porqué otorgar mención honorífica.			
Nombre jurados evaluadores	Andrés Felipe Giraldo Forero		
Firma	 FECHA: 23 de mayo de 2018		