



Instituto Tecnológico de Costa Rica

Ingeniería en Computadores

Estructuras y Algoritmos de Datos I

Anexo del Proyecto III

Profesor:

Isaac Ramirez

Estudiantes:

Valeria Morales Alvarado

2019052012

Sergio Ríos Campos

2019007977

Stephanie Villalta

2019068165

Ignacio Lorenzo

Fecha de entrega:

22 / 11 / 2019

II semestre 2019

Tabla de contenidos

Introducción.....	p.3
Diseño del proyecto.....	p.4
Historias de usuario.....	p.4
Diagrama de solución del problema.....	p.5
Trabajo en equipo.....	p.5
Primer plan de iteraciones.....	p.5
Segundo plan de iteraciones.....	p.6
Tercer plan de iteraciones.....	p.7
Cuarto plan de iteraciones.....	p.8
Bitácoras.....	p.9
Bitácora de Stephanie Villalta.....	p.9
Bitácora de Sergio Ríos.....	p.11
Bitácora de Valeria Morales.....	p.11
Bitácora de Ignacio Lorenzo.....	p.13

Introducción

Para el tercer proyecto del curso Estructuras y Algoritmos de Datos I de la carrera Ingeniería en Computadores se solicitó crear un REST API con formato JSON. Este tiene como propósito modelar grafos a partir de información ingresada por el usuario, el cual en el contexto propuesto serán las autoridades del OIJ como herramienta para la investigación de casos criminales. Los datos en este caso están centrados en la relación entre llamadas telefónicas, los individuos involucrados y la duración de las mismas. En este documento se analizarán los métodos utilizados para el desarrollo de dicha aplicación, así mismo la división de las tareas entre los integrantes del grupo y restricciones necesarias para llevar a cabo el trabajo.

Según los requerimientos impuestos en el instructivo se tenía como tareas principales: todo aquello relacionado al servidor REST API y formato JSON, la implementación de grafos en Java, junto a todas las funciones que este debe tener y la documentación completa.

Con respecto al grafo en sí, este debe tener tres partes fundamentales: nodo, que se refiere a los objetos que contienen los nombres de las personas involucradas en las llamadas; edges, que hace referencia a las conexiones, las cuales en este caso representan las llamadas y la duración de las mismas según la longitud; y el grafo, el cual contiene los elementos anteriores.

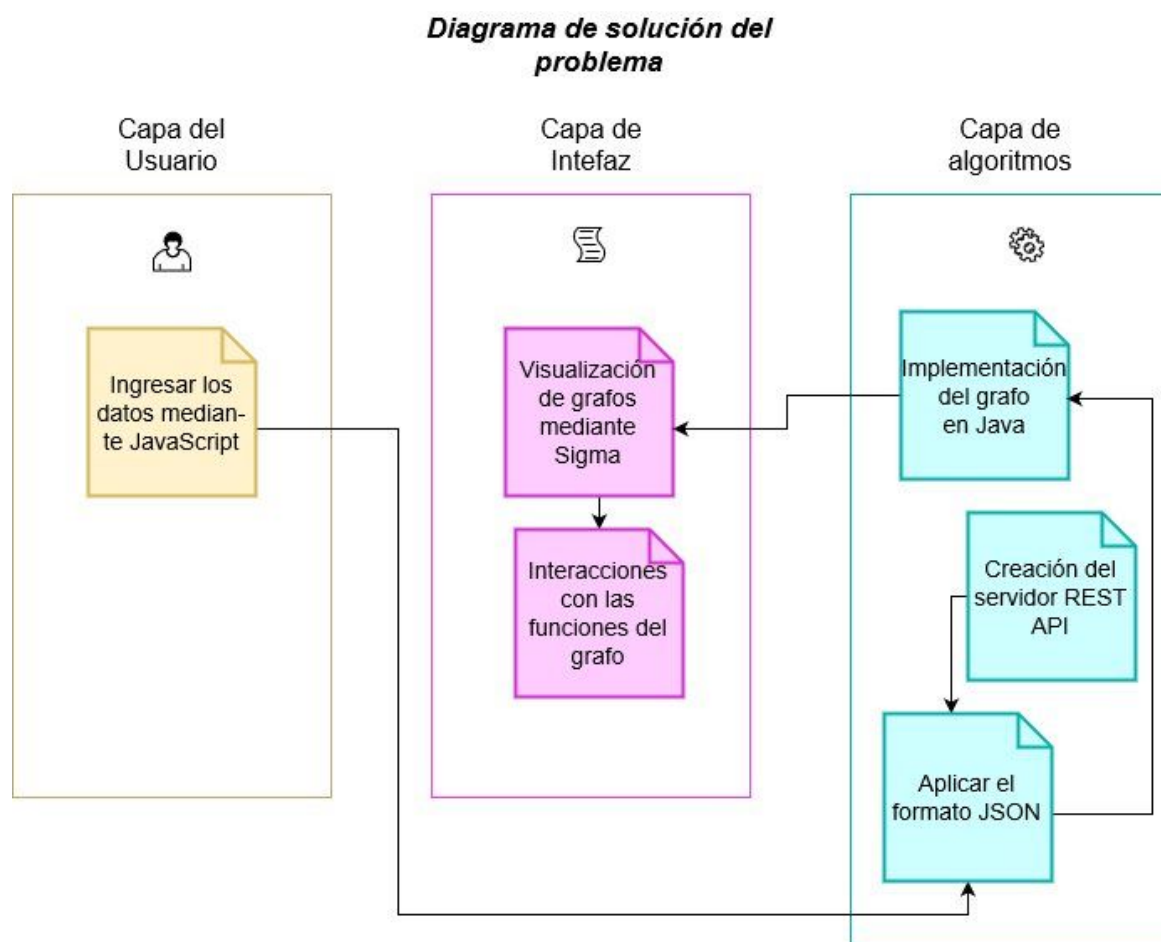
A partir de estos se dividió en tareas por medio de gitHub para que cada integrante pudiera avanzar por su cuenta en las distintas áreas del proyecto. Cada integrante se enfocó en una función específica del proyecto y así lograr optimizar el tiempo. Cabe destacar que en cada reunión de grupo se explicaban los avances y se trabajaba conjuntamente para solucionar errores y comprender el trabajo completo, así como ligar las partes.

Diseño del proyecto

Historias de Usuario:

1. yo como usuario/a quiero poder observar grafos para hacer debidamente los análisis criminológicos.
2. yo como usuario/a quiero poder importar un archivo CSV para construir un grafo que haga un análisis del contenido.
3. yo como usuario/a quiero que el grafo que se forme muestre los nodos como teléfonos con su id y el tamaño de enlace entre cada nodo, depende de la duración de la llamada y la conexión entre dos teléfonos.
4. yo como usuario/a , una vez que se muestran los grafos, quiero poder eliminar grafos para filtrar análisis.
5. yo como usuario/a, una vez creados los grafos, quiero poder agregar nodos para analizar datos previos con los nuevos.
6. yo como usuario/a quiero ser capaz de agregar y eliminar aristas para crear nuevos enlaces de análisis.
7. yo como usuario/a quiero ser capaz de actualizar aristas y grafos para poder más control de las posibles eventualidades que se den en el transcurso de la investigación.

Diagrama de arquitectura de la solución



Trabajo en equipo

Para el primer plan de iteraciones se buscó cumplir con las tareas más generales del proyecto, las propiedades que más destacan y dividir las entre los integrantes. Principalmente se basaba en investigar acerca del funcionamiento de aquellos formatos con los que no se contaba experiencia. Las tareas seleccionadas para el primer plan de iteraciones fueron:

- Investigar sobre Json y cómo implementarlo a la aplicación
- Conocer el funcionamiento de Postman

- Investigar sobre JavaScript
- Investigar sobre la creación de grafos
- Investigar sobre servidores con java (liberty)

Para este plan la división de tareas de cada integrante fue: para Valeria Morales se asignó la investigación de los grafos en Java, Stephanie e Ignacio se les asignó el servidor en java y la Wiki, Sergio se le asignó la parte de la interfaz del usuario y cambios al programa base hecho en Sigma para los grafos. Así mismo, todos los integrantes tuvieron que realizar una búsqueda general del funcionamiento de servidores y ciertas aplicaciones útiles para el desarrollo.

Las historias de usuario que se tomaron en cuenta para el primer plan fueron:

- yo como usuario/a quiero poder importar un archivo CSV para construir un grafo que haga un análisis del contenido.
- yo como usuario/a, una vez creados los grafos, quiero poder agregar nodos para analizar datos previos con los nuevos

En el segundo plan de iteraciones se continuó realizando investigación para ciertas funcionalidades y se inició la implementación de las partes del proyecto. Para esto se hizo un desglose de los requerimientos del proyecto y de las partes por completar previo a la unión de los trabajos realizados individualmente. Los objetivos propuestos fueron:

- Investigar cómo implementar REST API
- Conocer qué datos y rutas se necesitan para enviar y recibir datos del servidor
- Iniciar con la programación del funcionamiento de los grafos
- Implementar liberty a la aplicación para poder cerrar el servidor y usarlo con postman
- Iniciar con la programación del servidor
- Investigar cómo implementar Dijkstra

Para el segundo plan las responsabilidades para cada miembro fueron: Valeria Morales comenzó a implementar las funciones principales del grafo así como los

elementos principales y los atributos del mismo, Stephanie Villalta comenzó a probar implementaciones en el servidor con NodeJs y con Java, así como investigaciones sobre liberty con eclipseEE para iniciar con la programación del servidor haciendo pruebas con Postman, Ignacio investigó cómo implementar REST API y Dijkstra así como la manera de enviar y recibir datos del servidor y Sergio investigó sobre los servidores y como enviar datos de un CSV a Java.

Las historias de usuario que más poseen relación con el segundo plan de iteraciones son:

- yo como usuario/a quiero que el grafo que se forme muestre los nodos como teléfonos con su id y el tamaño de enlace entre cada nodo, depende de la duración de la llamada y la conexión entre dos teléfonos.
- yo como usuario/a, una vez creados los grafos, quiero poder agregar nodos para analizar datos previos con los nuevos.
- yo como usuario/a , una vez que se muestran los grafos, quiero poder eliminar grafos para filtrar análisis.

En el tercer plan de iteraciones se finalizó con la programación de variadas funciones para el grafo así como la creación del servidor. También se comenzó la unión de Java con el servidor para enviar funciones desde este y a continuación operar datos enviados desde la web. Las tareas de dicho plan fueron:

- Crear una forma para observar los datos y dar seguimiento a los cambios entre Java y Json
- Implementar Dijkstra
- Hacer pruebas con los programas unidos y editar programas de ser necesario
- Unir el programa de Java que contiene las funciones de los grafos con el REST API
- Conocer qué funciones tienen los grafos y el REST API

Para esta ocasión las labores se repartieron de la siguiente manera: Valeria Morales se enfocó en las funciones solicitadas para el grafo, como búsqueda de nodos, eliminación de los mismo, entre otras. Igualmente la relación que debe existir con el

servidor y las formas en las que se pueden recolectar los datos. Stephanie e Ignacio se encargaron de buscar la manera de conectar el servidor con los datos y el desarrollo de la Wiki, para este momento es importante recalcar que no se utilizó Nodejs como medio para el servidor, se procedió a utilizar liberty, java EE y sus respectivas librerías y Sergio hizo pruebas con servidores y Sigma.

Para el tercer plan de iteraciones se cumplió con los objetivos relacionados a las historias de usuario:

- yo como usuario/a quiero ser capaz de agregar y eliminar aristas para crear nuevos enlaces de análisis.
- yo como usuario/a quiero que el grafo que se forme muestre los nodos como teléfonos con su id y el tamaño de enlace entre cada nodo, depende de la duración de la llamada y la conexión entre dos teléfonos.
- yo como usuario/a , una vez que se muestran los grafos, quiero poder eliminar grafos para filtrar análisis.

Finalmente el cuarto plan de iteraciones busca conectar todas las partes para lograr el funcionamiento total de la aplicación. Además se debe detallar la interfaz gráfica junto con la conexión entre esta y el API. Las funciones como eliminar y buscar también se deben incorporar a la parte gráfica de la aplicación para que puedan ser retornadas a las funciones de Java y así conseguir hacer cambios al actualizar el grafo. Las labores por cumplir para este plan son:

- Sacar datos de CSV y usarlos para crear el grafo asociado
- Subir el archivo CSV por medio de la interfaz gráfica
- Crear la interfaz para el grafo
- Crear la interfaz gráfica del usuario
- Conectar mediante JSON el servidor a Java para generar un grafo nuevo
- Conectar los resultados de Java con Sigma para generar la interfaz del grafo

Los integrantes se dividieron los objetivos de la siguiente manera: Valeria se encargó de seguir desarrollando las funciones del grafo y recolectar datos JSON desde Java para generar el grafo. Por otra parte Stephanie se encargó de terminar la documentación e investigar sobre Sigma y sus implementaciones. Ignacio se

ocupó de conectar JSON a Java. Y Sergio conectó los resultados de Java a Sigma para generar la interfaz del grafo.

Es importante mencionar que en la transición del tercer plan al cuarto plan fue cuando surgieron la mayor cantidad de complicaciones. Debido a esto la mayoría de los integrantes trabajaron en buscar soluciones dichos errores.

Las historias de usuario relacionadas al trabajo realizado en el último plan de iteraciones fueron:

- yo como usuario/a quiero poder observar grafos para hacer debidamente los análisis criminológicos.
- yo como usuario/a quiero poder importar un archivo CSV para construir un grafo que haga un análisis del contenido.
- yo como usuario/a quiero ser capaz de actualizar aristas y grafos para poder más control de las posibles eventualidades que se den en el transcurso de la investigación.

Bitácoras

Stephanie Villalta Segura

- 29.10.2019

Investigación de Rest Api, JS, Java EE, java servers, Nodejs.

- 31.10.2019

Implementación de nodejs como prueba para la creación del servidor en java, utilizando JS. Configuraciones de nodejs para probar el programa.

- 1.11.2019

Inicio de la documentación.

- 4.11.2019

Reunión grupal, distribución de tareas.

- 5.11.2019

Investigación de Sigma js y Postman con servidores en java. Preparar los programas para iniciar. Instalar liberty en java, así como las librerías necesarias para seguir con el proyecto.

- 13.11.2019

Reunión grupal. Acomodo y cambios en la programación así como iniciar con la unión entre el servidor y los grafos, con los métodos y clases encargados de los nodos y edges de cada nodo. Por el momento metiendo datos a mano, no con archivo CSV.

Avances con la documentación, prácticamente lista, solo es de actualizarla.

- 20.11.2019

Investigación de los problemas recientemente surgidos en la programación. No se pueden agregar correctamente los grafos, ya que el programa no detenta lo que se le introduce. No se puede confirmar la funcionalidad del mismo por esta misma razón.

- 21.11.2019

Reunión grupal para poder unir correctamente los grafos con el server y ver más errores que se han desarrollado alrededor del desarrollo del API. Crear una lista en donde estén los datos y luego enviarla al Api de java por medio de una interfaz, esto porque se necesita ejecutar un método para crear los grafos y no se puede hacer desde JSON, entonces, se creará una lista con los archivos de JSON y luego se enviará a una clase en donde se sacarán los datos.

- 22.11.2019

Reunión grupal, análisis de problemas previos y soluciones logradas las cuales se mencionaron días atrás. Finalización de la documentación y Anexo del

proyecto(trabajo extra). Además, se siguió con la solución a los problemas entre las conexiones (java, server, interfaz gráfica), ya que todas están en archivos diferentes y su comunicación se ha complicado.

Sergio Ríos Campos

- **7-11-19**
Inicio de investigación sobre códigos JavaScript y protocolos HTTP
- **8-11-19**
Inicio de pruebas de códigos funcionales para subir un archivo .csv
- **10-11-19**
Código para subir archivo csv completado
- **10-11-19**
Inicio de investigaciones de conexión de la página web con el API creada
- **13-11-19**
Reunión con los integrantes del grupo para discutir aspectos del proyecto
- **14-11-19**
Conexión de la página web con API lograda
- **15-11-19**
Manejo de datos de csv dentro del API en Java
- **17-11-19**
Tratamiento de problemas con funciones del API
- **21-11-19**
Reunión con integrantes del grupo
- **22-11-19**
Reunión con integrantes del grupo

Valeria Morales Alvarado

5/11/2019

Se llevó a cabo la primera reunión grupal donde se discutieron las tareas principales del proyecto y se realizó la división de algunas labores iniciales, entre estas: generar el html, implementación de grafos, enviar la información al grafo, entre otras. Ese mismo día se implementaron las funciones principales del grafo, como por ejemplo: la clase Graph, GraphNode y Edge.

6/11/2019

Se inició la investigación sobre servidores y aplicaciones que serán necesarias para el desarrollo del proyecto, además se optimizaron algunas funciones del grafo.

13/11/2019

Se realizó la segunda reunión grupal donde se logró implementar el uso de llaves generadas de manera random mediante una variable UUID para reconocer y posteriormente realizar la búsqueda de nodos y edges con su respectivo índice. Además de esto se comenzó a desarrollar la función para eliminar nodos. Para la búsqueda de nodos y edges se creó un nuevo método en la clase graph que reciba una llave y recorra el los nodos o ejes del grafo, los cuales se encuentran en listas enlazadas hasta encontrar el objeto de igual identificación.

14/11/2019

Se corrigieron algunos bugs en el código del grafo relacionados al retorno de respuestas. De la misma manera se verificó que el grafo guardara los valores de los nodos y de los edges, así como sus respectivos atributos

20/11/2019

Se actualizó la función para eliminar nodos donde se consiguió que se eliminara el respectivo nodo del grafo y así mismo los edges que tienen relación al nodo eliminado, ya sea como origen o como destino.

21/11/2019

Se llevó a cabo la tercera reunión de equipo, en esta ocasión se tenía como objetivo unificar el trabajo realizado en JSON con la programación de los grafos en Java. Conjunto a esto se requería realizar arreglos de determinados bugs relacionados a la implementación de JSON. En relación a los grafos se incorporó una función que recolecta listas de tipo JSONArray y los convierta en nuevos elementos de un grafo, ya sean nodos o edges. También se hicieron avances en la documentación JavaDocs.

22/11/2019

Hubo una última reunión del equipo de trabajo en la cual se discutió acerca de las partes faltantes del proyecto. En este caso se buscó avanzar en la interfaz de los grafos y en la solución de bugs en el servidor. Durante la reunión también se realizó el anexo de la investigación y se continuó trabajando en los grafos. Posteriormente se buscó finalizar con la documentación y tratar de solucionar los errores al ingresar información desde el servidor a Java.

Ignacio Lorenzo

- **8/11/2019**

Se realizó algunas prácticas de Sigma JS para entender cómo funciona y como tiene que entrar los Nodos y edges para que se grafiquen y funcionen correctamente

- **9/11/2019**

Realice una investigación sobre el estilo de arquitectura REST y también sobre API, viendo ejemplos de códigos API que utilizan aplicaciones de grandes empresas, como Facebook entre otras.

- **13/11/2019**

Se inició las rutas para el API y todas su funciones Hubo un error al utilizar las Urls de los APIS ya que utilizaba funciones de otras URLS cuales no le corresponden y funciones tipo DELETE, PUT siendo la url específicamente para GET Se arreglo el

problema, con ayuda de códigos que hacen API y enseñan cómo montar un API en Java eclipse, entendiendo así cómo funcionan las Urls del API y GET...etc.

- **16/11/2019**

Se inició a implementar el grafo al API, cambiando la clase persona por un grafo con GraphNode y Edges, para luego poder graficarlo en la web con sigma JS y se empezó a modificar el API para que funcione correctamente con el grafo

- **18/11/2019**

Se añadió los errores de búsqueda para grafos con ID invalido y también para los Nodos y edges con ID invalido o que el ID no exista, esto para evitar que las funciones tengan error al buscar con un ID que no existe dentro de las funciones GET, PUT, DELETE.

Se encontró un error donde no se puede añadir un nuevo grafo desde postman (con el formato JSON) debido a como esta creado Graph, GraphNode, Edge, ya que primero se tienen que crear los GraphNodes para poder enlazar los Edge y luego utilizar un método de Graph para poder añadir los nodos y que el grafo no se encuentre vacío