



Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Visualización de Información

### **Proyecto 3 - Visualizaciones de Redes**

Realizado por:

Jennifer Alvarado Brenes c. 2020124171

Daniel Cornejo Granados c. 2019018538

Profesor:

Ing. Armando Arce Orozco

I Semestre 2024

## Contenido

<b>1. Introducción.....</b>	<b>3</b>
<b>2. Descripción del problema.....</b>	<b>3</b>
<b>3. Definición de fuentes de datos.....</b>	<b>4</b>
3.1. Diseases.....	4
3.2. Eurosis.....	4
3.3. School.....	4
<b>4. Descripción detallada.....</b>	<b>5</b>
4.1. Etapas del proyecto.....	5
4.1.1. Desarrollo de funciones serverless en Netlify.....	5
4.1.2. Creación de componentes en React.....	5
4.1.3. Pruebas de comunicación.....	5
4.1.4. Corrección de detalles.....	6
4.2. Decisiones de diseño.....	6
4.2.1. Menú de selección de interacciones.....	6
4.3. Pruebas del programa.....	6
4.3.1. Verificación de datos.....	6
4.3.2. Pruebas de representación gráfica.....	6
<b>5. Conclusiones.....</b>	<b>7</b>

## 1. Introducción

En el mundo actual, la visualización de datos juega un papel crucial en la comprensión de información compleja y en la toma de decisiones informadas. Este proyecto se centra en la creación de visualizaciones interactivas de redes, utilizando un modelo cliente-servidor. El objetivo principal es proporcionar una herramienta que permita a los usuarios explorar y analizar diferentes tipos de redes de datos, ofreciendo una representación clara y accesible de las relaciones y estructuras subyacentes.

Para lograr este objetivo, se han utilizado varias tecnologías modernas, destacándose:

- **Netlify:** Para la implementación de funciones serverless que procesan los datos y generan los modelos geométricos de las redes.
- **React:** Para la creación de componentes modulares que manejan la visualización e interacción con los grafos.
- **D3.js:** Para la representación gráfica de las redes utilizando la técnica de grafo de fuerzas (force-directed layout), que es especialmente adecuada para visualizar relaciones complejas entre nodos.

## 2. Descripción del problema

El objetivo de este proyecto es crear tres diferentes visualizaciones de redes utilizando un modelo cliente-servidor, en donde el servidor generará el modelo geométrico de la red y el cliente permitirá visualizar e interactuar con dicho modelo.

### Conjuntos de datos

Se deberá generar visualizaciones para cada uno de los conjuntos de datos que se detallan a continuación:

- **diseases:** consiste de la red de trastornos y genes patológicos vinculados por asociaciones conocidas entre genes y trastornos, lo que indica el origen genético común de muchas enfermedades. El conjunto de datos original puede consultarse como: The Human Disease Network, Goh K-I, Cusick ME, Valle D, Childs B, Vidal M, Barabási A-L (2007), Proc Natl Acad Sci USA 104:8685-8690.
- **eurosis:** Estudio topológico de la web EuroSiS: Mapeo de las interacciones entre los actores de Science in Society en la Web de 12 países europeos.

- **school:** Este conjunto de datos forma parte de un estudio de las redes de contacto en un centro de enseñanza primaria, del que se informa en el artículo High-Resolution Measurements of Face-to-Face Contact Patterns in a Primary School.

## **Generación del modelo geométrico**

Se deben crear tres funciones en Netlify que lean los anteriores archivos de datos y generen los respectivos modelos geométricos, utilizando D3.js, mediante la técnica de grafo de fuerzas (force-directed layout). Note que no se debe generar el SVG directamente en el servidor sino que eso será realizado por el cliente.

## **Visualización en el cliente (browser)**

Se debe crear una página web, utilizando D3.js, que permita visualizar cada uno de los archivos del modelo geométrico generados en Netlify. Esta visualización debe permitir consultar la etiqueta de cada nodo seleccionado, así como realizar acercamientos (zoom), alejamientos (unzoom), y corrimientos (pan).

# **3. Definición de fuentes de datos**

## **3.1. Diseases**

Este conjunto de datos describe la red de trastornos y genes patológicos vinculados por asociaciones conocidas entre genes y trastornos. Los datos muestran el origen genético común de muchas enfermedades y provienen del estudio "The Human Disease Network" por Goh K-I et al. (2007).

## **3.2. Eurosis**

Este conjunto de datos proviene de un estudio topológico de la web EuroSiS, mapeando las interacciones entre los actores de "Science in Society" en 12 países europeos. El objetivo es analizar cómo se relacionan y comunican estos actores a través de la web.

## **3.3. School**

Este conjunto de datos forma parte de un estudio sobre redes de contacto en un centro de enseñanza primaria, reportado en el artículo "High-Resolution Measurements of Face-to-Face Contact Patterns in a Primary School". Los datos se enfocan en los patrones de interacción cara a cara entre estudiantes y personal.

## **4. Descripción detallada**

### **4.1. Etapas del proyecto**

#### **4.1.1. Desarrollo de funciones serverless en Netlify**

La primera etapa del proyecto consistió en desarrollar funciones serverless alojadas en Netlify. Estas funciones son responsables de leer los archivos JSON proporcionados y generar los modelos geométricos de las redes utilizando la técnica de grafo de fuerzas (force-directed layout) de D3.js. El uso de Netlify permite escalar fácilmente las funciones y ofrece una infraestructura fiable para el procesamiento de los datos.

##### **Pasos involucrados:**

1. Configuración del entorno de desarrollo en Netlify.
2. Implementación de funciones para procesar los archivos JSON.
3. Generación de modelos geométricos utilizando D3.js.
4. Pruebas iniciales para asegurar la correcta ejecución y respuesta de las funciones.

#### **4.1.2. Creación de componentes en React**

La segunda etapa implicó la creación de componentes en React para cada uno de los grafos. Estos componentes manejan la lógica de visualización y permiten una fácil integración y reutilización dentro de la aplicación web.

##### **Pasos involucrados:**

1. Configuración del entorno de desarrollo en React.
2. Creación de componentes modulares para cada red.
3. Integración de D3.js para la visualización de los grafos.
4. Implementación de funcionalidades interactivas (zoom, pan, consultas).

#### **4.1.3. Pruebas de comunicación**

Se realizaron pruebas exhaustivas para asegurar la correcta comunicación entre la página web y las funciones serverless en Netlify. Estas pruebas fueron cruciales para garantizar que los datos se transmitieran adecuadamente y que las visualizaciones respondieran como se esperaba.

##### **Pasos involucrados:**

1. Verificación de la correcta transmisión de datos desde Netlify a la aplicación React.
2. Pruebas de latencia y rendimiento de las funciones serverless.

3. Ajustes y corrección de errores identificados durante las pruebas.

#### **4.1.4. Corrección de detalles**

Tras la etapa de pruebas, se identificaron y corrigieron detalles tanto en las funciones serverless como en los componentes React. Estas correcciones optimizaron el rendimiento y mejoraron la experiencia del usuario.

#### **Pasos involucrados:**

1. Revisión de código y optimización de funciones.
2. Mejora en la gestión de estados y propiedades en React.
3. Ajustes en la visualización y manejo de eventos interactivos en D3.js.

### **4.2. Decisiones de diseño**

#### **4.2.1. Menú de selección de interacciones**

Una de las decisiones de diseño más importantes fue implementar un menú de selección para el tipo de interacción deseada. Al intentar combinar todas las interacciones (consulta de etiquetas, zoom, pan) simultáneamente, se encontró que la experiencia de usuario se volvía confusa y poco intuitiva. Separar las interacciones en un menú permitió una navegación más fluida y comprensible.

#### **Razonamiento detrás de la decisión:**

- Facilitar el uso y la comprensión de las herramientas interactivas.
- Reducir la complejidad de la interfaz.
- Mejorar la experiencia del usuario al proporcionar controles claros y específicos para cada tipo de interacción.

### **4.3. Pruebas del programa**

#### **4.3.1. Verificación de datos**

Se llevó a cabo una verificación exhaustiva de los datos enviados desde las funciones serverless para asegurar su corrección y completitud. Esta verificación garantizó que los modelos geométricos se generaran adecuadamente y que la información presentada en la visualización fuera precisa.

#### **Métodos utilizados:**

- Comparación de datos procesados con los archivos JSON originales.
- Verificación de integridad y formato de los datos.

#### **4.3.2. Pruebas de representación gráfica**

Se realizaron pruebas para asegurar que los nodos y los enlaces de las redes se representaran correctamente en la visualización. Esto incluyó la comprobación de que las interacciones (consulta de etiquetas, zoom, pan) funcionaran sin problemas y de manera eficiente.

#### **Métodos utilizados:**

- Inspección visual de las redes generadas.
- Pruebas de interacción manuales para verificar la funcionalidad de zoom, pan y consultas.
- Ajustes en el layout de fuerzas para mejorar la claridad y presentación de las redes.

## **5. Conclusiones**

El proyecto logró con éxito su objetivo de crear una plataforma interactiva para la visualización de redes complejas. Utilizando funciones serverless en Netlify y componentes en React, se logró una visualización eficiente y fácil de usar. La implementación de un menú de selección para las interacciones mejoró significativamente la experiencia del usuario, permitiendo una navegación clara y fluida.

Las pruebas exhaustivas aseguraron que tanto la comunicación entre cliente y servidor como la representación gráfica fueran correctas y funcionales. Los resultados muestran una herramienta robusta y versátil para explorar y analizar redes de datos complejas, proporcionando una base sólida para futuras expansiones y mejoras.

El proyecto no solo demuestra la capacidad de integrar tecnologías modernas como Netlify, React y D3.js, sino que también resalta la importancia de un diseño centrado en el usuario para aplicaciones de visualización de datos. En conclusión, la plataforma desarrollada es una contribución valiosa para el análisis y la visualización de redes complejas, ofreciendo una interfaz intuitiva y poderosa para investigadores y usuarios finales.

En el siguiente link, se puede encontrar la versión final del proyecto:  
<https://viz-project-3.netlify.app/>