

## Análisis de Tecnologías Web OpenSource para la Visualización de Datos Aplicando Grafos

Luis-Alberto Jumbo- Flores <sup>a</sup>, Pablo Quezada<sup>a, b</sup>

<sup>a</sup> Universidad Tecnica Particular de Loja 1101608,  
San Cayetano Alto, Ecuador  
{ljumbo, paquezada, }@utpl.edu.ec  
www.utpl.edu.ec

<sup>b</sup> Technical University of Madrid  
Departamento de Sistemas Informaticos CITSEM,  
Ctra. De Valencia, Km. 7. E-28031, Madrid, Spain  
pablo.quezada.sarmiento@alumnos.upm.es  
www.upm.es

**Resumen:** Hoy en día se puede encontrar diversas herramientas de software para la visualización de datos como: paquetes de software, plataforma de desarrollo, y específicamente *Apis*. Es por ello que se hace necesario la investigación y evaluación de dichas herramientas, con finalidad de disponer de un punto de partida cuando se desea construir soluciones que permitan mostrar información de manera gráfica. El presente paper se enfoca en desarrollar una Revisión Sistemática sobre las tecnologías *OpenSource* para la visualización de datos, aplicando grafos; así mismo, el establecimiento y empleo de mecanismos para la evaluación de tecnologías OpenSource para la visualización de datos en plataformas Web.

**Palabras clave:** Tecnologías Web, OpenSource, Visualización, Grafos, Graph Drawing.

**1. Introducción:** En la actualidad la gran cantidad de datos que existen tanto en la web como en la variedad de sistemas transaccionales, hacen necesario la construcción de: utilitarios, librerías, *Apis*, incluso aplicaciones, que permitan explotar de una forma eficiente la información de dichos datos. En consecuencia, y como es de suponer, las formas de mostrar la información a los que toman decisiones (usuario finales, gerentes, directores de proyectos, inversionistas, etc.) , ya no solo se limitan a mostrar informes escritos, con tablas de datos y alguno que otro gráfico estadístico. Ahora estamos en una nueva era, la era en donde la información debe incluir características como la comunicación visual.

Pero en si ¿qué es la visualización de datos? Para argumentar la definición primero pensemos que los datos son diamantes en bruto, a simple vista los datos son oscuros (difíciles de comprender), sin embargo después de ser sometidos a un proceso de limpieza y algoritmos de visualización podemos apreciar lo claro que resulta la información (un diamante pulido en todo su esplendor).

La visualización de datos es el uso de tecnologías que permiten representar los datos sometidos a un proceso de análisis y limpieza, para mostrar información gráficamente. La visualización permite representar la información de múltiples formas: Gráficos estadísticos, Gráficos jerárquicos, Gráficos de Redes, Gráficos de Grafos., Gráficos de dispersión, entre otros.

El presente paper tiene como objetivos desarrollar una Revisión Sistemática sobre las tecnologías *OpenSource* para la visualización de datos; así mismo establecer y emplear mecanismos para la evaluación de Tecnologías para la visualización de datos *OpenSource* en la web.

**2. Trabajos relacionados:** Existen trabajos que abordan los grafos en la web, pero estos trabajos son específicos en la solución de problemas puntuales como en [1], en este trabajo de usan los grafos para identificar clústeres, a través de la definición de algoritmos. Según [2], se puede construir una representación gráfica del ciberespacio, a través los diferentes hipervínculos de las páginas web. Otro estudios como [3], ofrecen nuevos algoritmos de distribución de nodos dentro de un grafo, a través de algoritmos basados en fuerza denominados force-Directed Drawing Method.

**3. Metodología de Investigación:** En 2004-2005, Kitchenham, Dyba y Jørgensen escribieron tres artículos que sugieren que el concepto de práctica basada en la evidencia, que se puede adoptar en ingeniería de software [8]. Según [9] los estudios de mapeo utilizan la misma metodología básica de la Revisión Sistemática de la Literatura (SLR), pero tienen como objetivo identificar y clasificar toda la investigación relacionada con un tema de ingeniería de software.

En esta sección se describe y discute la metodología de investigación utilizada e impulsada por un mapeo sistemático, que incluye la técnica de recolección de datos y el método de análisis de datos. Se realizaron estudios de mapeo sistemático para evaluar la cantidad y tipos de estudios primarios en un área de interés de una manera imparcial y sistemática. [10]. El estudio de mapeo sistemático es una técnica en la Evidencia Basada en Ingeniería de Software y ya ha sido utilizado en diferentes subdominios de la ingeniería de software es decir, pruebas de software y especificación de requisitos [11], [12]. El mapeo sistemático requiere una buena planificación, ejecución y análisis sistemático.

Para el desarrollo de la estrategia de búsqueda fue necesario considerar el título, resumen y palabras clave de los artículos en las bases de datos electrónicas incluidas y actas de congresos se buscará acuerdo con la siguiente estrategia de búsqueda [13].

- 1) Palabras claves.
- 2) Variantes y acrónimos para Grandes Términos: Cadena de búsqueda: [14].

Para llevar a cabo el mapeo sistemático fue necesario utilizar una estrategia de búsqueda formal a través de prácticas recomendables en contextos de investigación como la respetabilidad y las revisiones externas de esta contribución [15], [16]. En un intento de llevar a cabo una búsqueda exhaustiva identificamos cinco fuentes electrónicas de relevancia para la Ingeniería de Software (Tabla 1).

**Tabla 1: Bases de datos electrónicas consultadas.**

Bases Electrónicas		
Id	Base de Datos	Enlace
1	IEEE Xplore	<a href="http://www.ieeexplore.ieee.org">www.ieeexplore.ieee.org</a>
2	ACM Digital Library	<a href="http://www.dl.acm.org">www.dl.acm.org</a>
3	ISI Web of Knowledge	<a href="http://www.webofknowledge.com">www.webofknowledge.com</a>
4	Science Direct	<a href="http://www.sciencedirect.com">www.sciencedirect.com</a>
5	Springer	<a href="http://www.springer.com">www.springer.com</a>

En el mismo contexto se utilizaron criterios de selección del estudio que pretendieron identificar los estudios primarios que proporcionan evidencia directa acerca del tema de investigación. A fin de reducir la probabilidad de sesgo, los criterios de selección deben ser decididos durante la definición de protocolo, aunque pueden ser refinados durante el proceso de búsqueda [9], [12]. La selección de los estudios es un proceso de varias etapas. Inicialmente, los criterios de selección deben interpretarse liberalmente, por lo que a menos que un estudio identificado por las búsquedas electrónicas y manuales puede excluirse claramente basado en el título y el resumen [12]. Una parte clave del mapeo sistemático es la extracción de datos, en la que se obtienen de texto y datos esenciales de los estudios primarios de una manera explícita y consistente de acuerdo a una estrategia de extracción definido. Sin embargo, antes de iniciar la extracción, se recomienda la lectura de todo el conjunto de trabajos seleccionados al menos una vez [4]. Los documentos [8], [10], [17] sugieren:

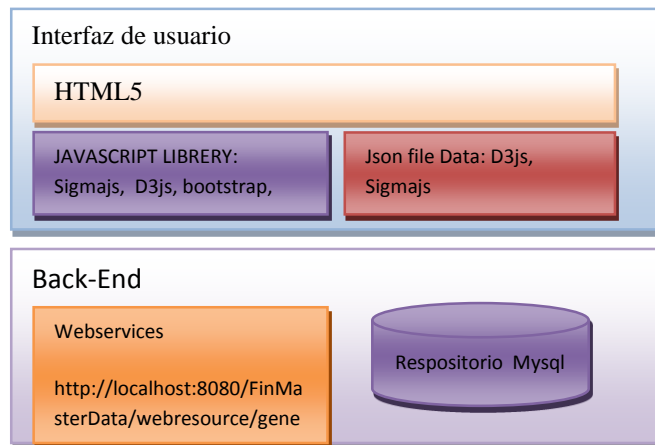
- Datos de la publicación (por ejemplo, los autores, año, título, fuente, abstracto, tiene como objetivo); descripciones de contexto (por ejemplo, temas, tecnologías, la industria, los ajustes).
- Hallazgos.

Según [8], en un mapeo sistemático, es fundamental para identificar, extraer y sintetizar correctamente la información de los estudios incluidos para proporcionar respuestas fiables a la investigación en este caso enfocado a la evaluación de datos aplicando grafos. Una vez que los datos de los estudios integrales han sido extraídos correctamente, han sido sintetizados a fin de proporcionar nuevos conocimientos en las áreas de investigación. En el mismo contexto, si la información está disponible a partir de estudios en curso, se debe incluir el suministro de información de calidad adecuada sobre el estudio se puede obtener un permiso por escrito y está disponible de los investigadores [10].

Con el fin de desarrollar una síntesis de los datos, fue necesario tener en cuenta la identificación de cada estudio primario, el nombre de los autores, el año de publicación, los criterios de investigación utilizados en cada base de datos se describe en la tabla 1, los resultados de acuerdo a cada pregunta de investigación y la puntuación de acuerdo con los criterios de calidad para evaluar los estudios primarios. Los 11 criterios relacionados con la calidad que se tomaron en cuenta al evaluar los estudios identificados en la revisión sistemática.

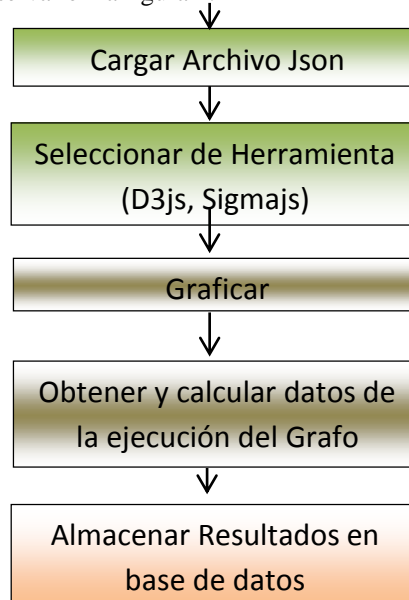


Evaluación de Herramientas: Para continuar con la evaluación, se desarrolló un prototipo que implementa las tecnologías seleccionadas considerando los puntajes más altos: D3js y Sigmajs. La arquitectura del prototipo se muestra a continuación:



**Fig. 1.** Arquitectura del Prototipo

Como resultado de la arquitectura de software planteada previamente se obtiene el respectivo prototipo; que describe un proceso general guiando la ejecución del prototipo, lo cual se puede observar en la figura 2.



**Fig. 2:** Diagrama de Flujo General

Presentación de resultados: Para determinar los resultados que permitan evaluar las herramientas se debe ejecutar el prototipo (Figura 3) a través del cargado de archivos de grafos en formato Json de acuerdo al proceso de la figura 2. Para tal fin se estableció los archivos de la tabla 4.

**Tabla 4.** Archivos Json para Evaluar la Tecnologías

Archivo Sigmajs	Archivo D3js
dato1.sigmajs.json	dato1.d3.json
JsonGraph.sigmajs.json	JsonGraph.d3.json
misera.sigmajs.json	misera.d3.json
JsonOtro.sigmajs.json	JsonOtro.d3.json

The screenshot shows a web application titled 'Evaluador de Herramientas para la Visualización de datos Aplicando Grafos'. The interface includes a sidebar with the following sections:

- Cargar el Archivo a visualizar:** A text input field containing 'dato1.s...a.json'.
- Datos del archivo:** A JSON snippet showing node information:
 

```
{
  "nodes": [
    {
      "id": "1",
      "label": "Afiliación_Universidad_Tecni"
    }
  ]
}
```
- Número de Ejecuciones:** A text input field containing '40'.
- Selección Librería a usar:** A dropdown menu with 'SigmaJS' selected.

The main content area features a 'Tabla de datos' (Data Table) with the following columns: Id, Herramienta, Tipo de Grafo, # Nodos, # Enlaces, # Max. Enlaces, Densidad Grafo, and Ejecución. The table contains six rows of data:

Id	Herramienta	Tipo de Grafo	# Nodos	# Enlaces	# Max. Enlaces	Densidad Grafo	Ejecución
1	sigmajs	Pequeño	48	60	1128	0.0532	99
2	sigmajs	Pequeño	48	60	1128	0.0532	18
3	sigmajs	Pequeño	48	60	1128	0.0532	16
4	sigmajs	Pequeño	48	60	1128	0.0532	12
5	sigmajs	Pequeño	48	60	1128	0.0532	13
6	sigmajs	Pequeño	48	60	1128	0.0532	8

**Fig. 3:** Ejecución del prototipo.

#### 4. Marco teórico:

En el siglo XVIII, la ciudad de Königsberg, en Prusia Oriental, acostumbrar la población a dar paseo dominicales sobre siete (7) puentes que conectaba a cuatro (4) zonas de la ciudad. La tarea era recorrer dichos puentes sin tener que pasar por ellos más de dos veces, y tratar de llegar al punto de partida. Euler propuso un grafo, que representa las zonas de la ciudad como nodos, y las aristas fueron expuestas como aristas.

Para comprender la visualización de datos primero debemos entender el concepto de Graph Drawing, que es una combinación de las matemáticas y ciencias de la computación que usan métodos de teoría de grafos geométricos, y visualización de información, con la finalidad de obtener representaciones graficas bidimensionales, derivadas de análisis de redes, cartográficas, etc.

Básicamente *Graph Drawing* ofrece posibilidades de presentar información partiendo de nodos y aristas de un grafo; en incorporados métodos de distribución de nodos llamados *methods layouts*.

Según [4] matemáticamente un grafo  $G$  constan de un conjunto de *vértices*  $V(G)$  o *nodos*  $N(G)$  y un conjunto *enlaces*  $E(G)$  o *aristas*  $A(G)$ , que une un vértice con otro.

$$G = (V, E) = (N, A)$$

El número de nodos de grafo se llama *orden* de grafo y se expresa como:

$$ord(G) = |V(G)| = |N(G)|$$

El número de enlaces o aristas se llama *tamaño* del grafo y se expresa como:

$$tam(G) = |E(G)| = |A(G)|$$

Dos vértices se dicen *adyacentes* si existen una arista formada por esos vértices, en caso contrario se denominarían *disjuntos*

Una arista es *incidente* si comparte un vértice.

El *grado* es el número de vértices adyacentes, si un vértice tiene orden cero se denomina *aislado*

El *grado* de un vértice  $x$  es número de adyacentes a él, esto es, el número de aristas incidentes a  $x$  y se denota por

$$grad(x)$$

A modo de definición un grafo es un gráfico compuesto de nodos o vértices y enlaces o aristas, que pueden estar o no dirigidos.

#### 4.1 Características de un grafo

Los vértices (nodos) se suelen representar por círculos, pero también se pueden representar por otras figuras geométricas como: triángulo, cuadrado, etc.

Los enlaces (aristas) se suelen representar por líneas: rectas y/o curvas.

Tanto los nodos y aristas pueden asignarse valores que se toman como identificadores, y/o pesos.

Las aristas puede que actúan como enlaces y pueden o no tener flechas que indican una dirección, según sean grafos dirigidos y no dirigidos.

## 4.2 Medidas de los grafos

Cuando se aplica grafos para representar y/o analizar datos, es común encontrar medidas o métricas que permite evaluar el grafo en cuestión:

**Número de Cruces:** el número de aristas que llegan y salen de un determinado nodo.

**Área del grafo:** es el tamaño más pequeño de un cuadro que delimita en el área del grafo, básicamente se toma como referencia las distancias entre dos nodos.

**Visualización de Simetría:** un problema frecuente es tratar de encontrar grupos de simetría, dentro de un grafo. Estos grupos de simetría son nodos interconectados pero que de alguna forma están organizados por sus relaciones y algoritmos de distribución de nodos.

## 4.3 Método de Disposición de nodos (Node layouts methods).

Existen múltiples tipos de métodos *layout* que permite distribuir los nodos en un grafo.

*Layout* basado en fuerza (*force-based layout*) son algoritmos que modifican continuamente la posición inicial de los nodos, a través de la simulación de fenómenos físicos, relacionados a fuerzas de atracción o repulsión entre nodos adyacentes.

**Layout de algoritmos de Árboles:** Similar a como se grafican los árboles en estructura de datos tipo árbol en estructuras de datos. Son útiles para describir estructuras jerárquicas.

## 4.4 D3JS Data-Driven Documents

Según [5] y [6], se define con una librería java script para manipular datos y producir visualizaciones dinámicas en navegadores web, a través del uso de estándares como: HTML5, SVG, y CSS. D3js funciona en casa la mayoría de los navegadores Web modernos, a excepción de IE8 y las versiones anteriores.

*“El nombre d3 tiene su significado en DATA-DRIVEN DOCUMENTS. Mike Bostock en el 2011 desarrollo una librería que permite dibujar gráficos mediante el empleo de datos, sin preocuparse por a disposición de las formas gráficas; pues la librería se encarga de encajar todas las formas dentro del grafico” [18]*



#### 4.5 Sigmajs

Sigmajs es una librería javascript que permite desplegar grafos, y está diseñada como un motor personalizable y escalable para construir aplicaciones web interactivas que requieren visualizar grafos [7].

#### 5. Resultados:

Considerando que cada archivo definido en la tabla 4, se sometió a la ejecución del prototipo, se obtuvo los datos que fueron sometidos a una evaluación estadística, se muestra la evaluación de una de las pruebas, la información se puede apreciar en la tabla 5.

Tabla 5. Datos del archivo

Nombre del archivos:	dato1.d3.json, dato1.sigma.json	
Nodos		48
Enlaces		60
Tipo de grafo		Pequeño

En la figura 7 se muestra el número de ejecuciones en el eje x y la cantidad de milisegundo necesarios para la visualización del grafo.

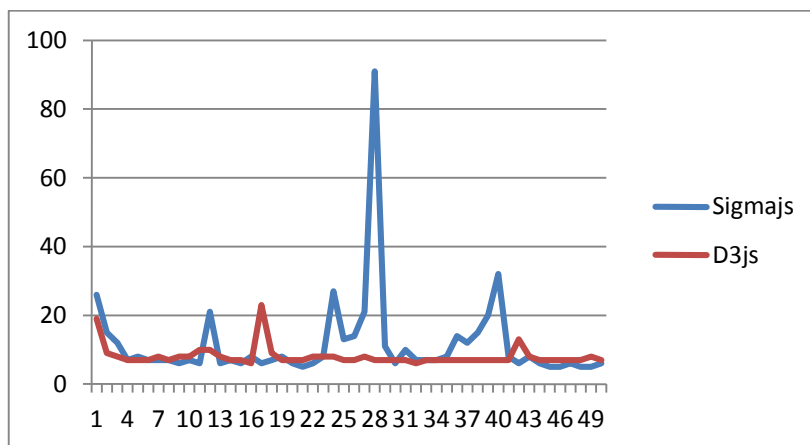


Fig.7. Grafica estadísticas de ejecución

En la tabla 6 se muestra el resumen de datos tiempo mínimo, máximo promedio así una interpretación de estos resultados.

**Tabla 6.** Resumen de resultado

Herramienta	Tiempo Mínimo	Tiempo Máximo	Tiempo Promedio
SIGMAJS	5	91	11,54
D3JS	6	23	8.06
Observaciones:  Como podemos apreciar sigmajS tiene el tiempo más pequeño entre los tiempos mínimos, sin embargo d3js posee el tiempo más pequeño entre los tiempo máximos, además como podemos observar el tiempo promedio de d3 es inferior a sigma en razón de 0,6984			

Para complementar la evaluación de las Herramientas Web OpenSource (D3, JS, Sigma JS) se hace necesario establecer parámetros adicionales de evaluación como facilidad de uso, documentación y liberaciones tal como se puede apreciar en la tabla 7. La aplicación de esta tabla sobre las tecnologías mencionadas se muestra en la tabla 8.

**Tabla 7.** Parámetro y valoraciones

<b>Parámetro</b>	<b>Descripción de la valoración</b>	<b>Descripción de valores a asignar</b>
Facilidad de uso	Valoración dada por la facilidad de construir funciones de la herramientas	1: No es fácil 2: Regularmente fácil 3: Fácil 4: Muy fácil
Documentación	Valora la disponibilidad y actualización, de la documentación de las herramientas.	1: No Disponible 2: Disponible y No Actualizada 3: Disponible y Actualizada
Liberaciones	Valora la cantidad y frecuencia de la liberaciones de la Librería	1: Pocas liberaciones 2: Algunas liberaciones 3: Variadas liberaciones 4: Muchas liberaciones

**Tabla 8.** Resultados Evaluación:

<b>Herramienta</b>	<b>Facilidad de Uso</b>	<b>Documentación</b>	<b>Liberaciones</b>
SigmaJS	4	3	3
D3JS	3	3	4

## 6. Conclusiones:

Los resultados del documento presentan los criterios para evaluar las herramientas open Source para la de datos aplicando grafos.

- Existen diversidad de herramientas para la visualización de datos aplicando grafos bajo plataforma web tales como: Processingjs, Raphaëljs, D3js, Sigmajs, etc.
- Tanto Sigmajs como d3Js son útiles para la visualización de grafos en aplicaciones web, y además son herramientas de uso libre.
- D3 muestra un dominio superior en cuanto a la ejecución, pues en cada una de las pruebas realizadas su media de ejecución es por debajo de la Sigmajs, a pesar que los tiempos mínimos de sigma fueron siempre los menores.
- A pesar de la superioridad de d3js sobre sigma en los tiempos de ejecución como se muestran en la graficas estadísticas, debemos indicar que la facilidad de uso de sigma compensa en algo el dominio de D3js.
- La documentación es utilizable y actualizada tanto en Sigmajs como en d3 y contantemente está actualizándose.
- La cantidad de liberaciones de d3js es superior con respecto a sigma, se puede apreciar en las versiones de cada una de las herramientas.

## Agradecimientos:

Los autores desean expresar sus agradecimientos a la Universidad de Alcalá de Henares, Universidad Politecnica de Madrid (UPM) en especial al Programa de Doctorado de Ciencias y Tecnologías de la Computaciòn para Smart Cites, al Departamento de Sistemas Informaticos de la UPM, a la Universidad Tecnica Particular de Loja, Departamento de Ciencias de la Computaciòn y Electrónica, Vicerrectorado de Investigación UTPL así como a los organizadores del CACIED 2015 por el constante apoyo brindado.

## Referencias:

1. Xiaodi H., Wei L.: Identification of Clusters in the Web Graph Base on Link Topology, IEEE Xplore Digital Library, (2003)
2. Xiaodi H., Wei L., Di Z., Maoulin H., Quang V. Nguyen: A Kernel-Based Algoritihm for Multilevel Drawing Web Graph, IEEE Xplore Digital Library, (2007)
3. Hiroki O.;Kozo S.: Force-Directed Drawing Method for Intersecting Clustered Graoah, (2007)
4. Caicedo A., De Garcia G., Mendez R.: Introducción a la Teoría de Grafos, 1rd edn. Ediciones Elizcom, (2010)
5. D3.js, Wikipedia EN, <http://en.wikipedia.org/wiki/D3.js>, (20/07/2015)
6. Github, Home D3js, <https://github.com/mbostock/d3/wiki>, (20/07/2015)
7. Jacomy A., GitHub Sigmajs. <https://github.com/jacomyal/sigma.js/wiki/Graph-API> (27/06/2015)
8. Kitchenham, B., Budgen, Brereton., & Pearl D., (2011), "Using mapping studies as the basis for further research—a participant-observer case study.*Information and Software Technology*", 53(6), 638-651, Science Direct.
9. Cruzes,D. & Tore D., (2011), " *Research synthesis in software engineering: A tertiary study. Inf. Softw. Technol*", 53(5):440455
10. Kitchenham, B., & Charters, S., (2007), " *Guidelines for performing Systematic Literature Reviews in Software Engineering*", Technical Report, EBSE, IEEE.
11. R. S. M. Petersen, K.; Feldt & Mattsson, M. (2008), " *Systematic mapping studies in software engineering*" In *EASE'08*, pages 1–10, Bari, Italy.
12. Fernandez, C., Daneva, M.; Sikkel, K.; Wieringa, R., Dieste, O. & Pastor, O. (2009). " *A systematic mapping study on empirical evaluation of software requirements specifications techniques*", ESEM 3rd International Symposium, 502-505.
13. Mannan, M. & Usan, M., (2011), " *Software Engineering Curriculum: " A Systematic Mapping Study"*", IEEE.

14. Brereton, P., Kitchenham, B., Budgen, D. & Turner, M., (2007), "*Lessons from applying the systematic literature review process within the software engineering domain.*", JSS 80, pp. 571-583.
15. Kitchenham, B., Pfleeger, S., Pickard, L., Jones, P. & Rosenberg, J., (2002), "*Preliminary guidelines for empirical research in software engineering*", IEEE, Transactions on Software Engineering.
16. Greenhalgh, T., (2001), "*How to Read a Paper*", Second ed., BMJ Publishing Group, London.
17. Shaw, M., Herbsleb, J., Ozkaya, I. & root, D., (2006), "*Deciding What to Design: Closing a Gap in Software Engineering Education*", Springer-Verlag Berlin Heidelberg.
18. Teller, S. (2013). *Data Visualization with D3.js*. Birmingham: Packt Publishing Ltd.