

# WebGL Based 3D Dashboard for Tracking Software Development

Adrián Alonso Barriuso

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN  
Universidad Rey Juan Carlos

2016



# Contenido

- 1 Introducción
  - Descripción del problema
  - Objetivo principal
- 2 Tecnologías utilizadas
- 3 Desarrollo
  - Metodología
  - Iteraciones
- 4 Diseño y resultados
  - Arquitectura
- 5 Conclusiones
- 6 Referencias



## Dashboards y seguimiento de desarrollo de software

Un **dashboard** es una interfaz de usuario que nos permite manejar y gestionar un tipo determinado de software o de hardware. En nuestro caso trabajamos con dashboards de visualización de datos de desarrollo de software. Algunos ejemplos de dashboards y de librerías para la visualización de datos basadas en software libre son:

- **Kibana**
- **Freeboard**
- **dc**



# Kibana

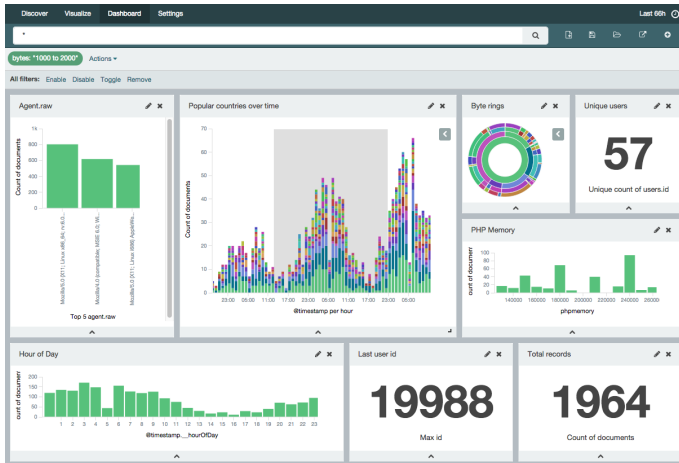


Figura: ejemplo de Kibana



# Freeboard



Figura: Ejemplo de freeboard



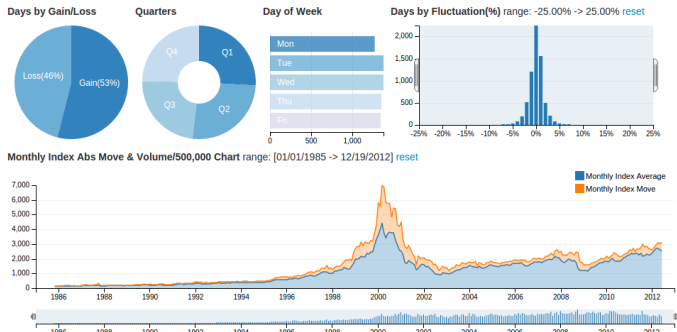


Figura: ejemplo de uso de dc.js

## Dashboards en 2D

### Ventajas

- 1 Simplicidad de representación.
- 2 No requieren aceleración gráfica.
- 3 Impacto mínimo en el rendimiento.

### Inconvenientes

- 1 No se pueden representar gráficos con mas de 2 dimensiones, por razones obvias.
- 2 Las posibilidades de visualización y colocación son limitadas.



## Objetivo principal

Crear una librería que nos permita crear visualizaciones y filtrar datos de desarrollo de software en 3D, dentro de cualquier navegador.

### Requisitos

- 1 Conseguir una interfaz y funcionalidad tan parecidos a los de dc.js como sea posible.
- 2 Tener varios tipos de gráficas y ser capaces de filtrar.
- 3 Utilizar un framework de WebGL
- 4 Respuesta rápida de los filtros.
- 5 Debemos ser capaces de hacer zoom, desplazarnos y arrastrar las gráficas.
- 6 Debemos poder colocar gráficas tanto de forma independiente como dentro de paneles.
- 7 Tener una estructura basada en programación orientada a objetos.





## Tecnologías utilizadas

- HTML5
- Javascript
- **webGL**
- **Three.js**
- Crossfilter
- THREEEx.DomEvents
- Orbit controls
- dat.GUI



# Metodología Scrum

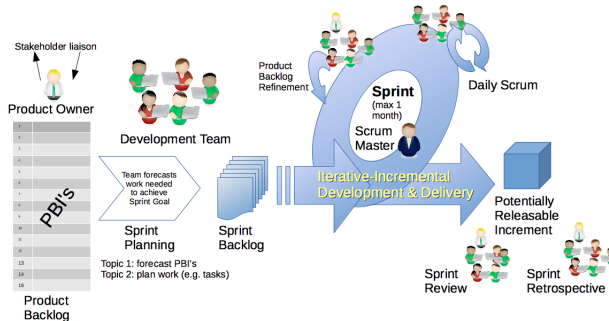


Figura: Metodología Scrum

# Iteraciones

- **Iteración 0:** Investigación y aprendizaje
- **Iteración 1:** Primeras demos
- **Iteración 2:** Añadir interactividad
- **Iteración 3:** Añadir posibilidades de filtrado dimensional.
- **Iteración 4:** Creación de la biblioteca
- **Iteración:** Añadir paneles y otras características.



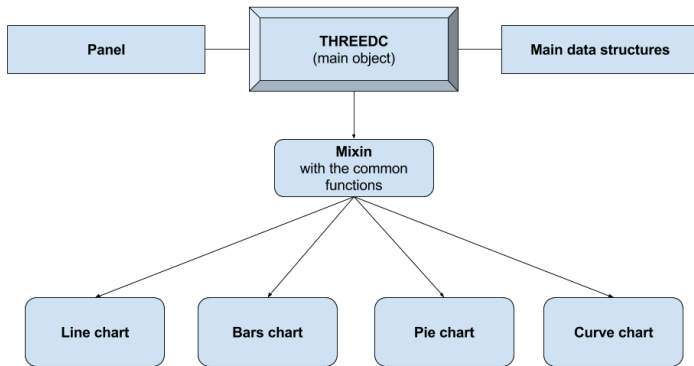


Figura: Arquitectura de la librería

## Referencias



A.J. Chorin, J.E. Marsden.

*A Mathematical Introduction to Fluid Mechanics*  
Springer-Verlag, 1980.



K. Devlin.

*The Millenium Problems. The Seven Greatest Unsolved Mathematical Puzzles of Our Time*  
Basic Books, 2002.



C. Fefferman.

*Clay Mathematics Institute, Millenium Problems. Official problem description.*

[http://www.claymath.org/millennium/Navier-Stokes\\_Equation](http://www.claymath.org/millennium/Navier-Stokes_Equation)



Wikipedia contributors

*Navier-Stokes equations*

Wikipedia, The Free Encyclopedia., 2008.

[http://en.wikipedia.org/wiki/Navier-Stokes\\_equations](http://en.wikipedia.org/wiki/Navier-Stokes_equations)

