UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

TFG del Grado en Ingeniería Informática

Aplicación web para la visualización y búsqueda de datos almacenados en dispositivo con linux embebido.

Presentado por Rodrigo Martínez Bravo
en Universidad de Burgos — 12 de diciembre de 2017
Tutor: Jesús Enrique Sierra García

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

D. Jesús Enrique Sierra García, profesor del departamento de Ingeniería Civil, área de lenguajes y sistemas informáticos.

Expone:

Que el alumno D. Rodrigo Martínez Bravo, con DNI 71298529R, ha realizado el Trabajo Final de Grado en Ingeniería Informática titulado: Aplicación web para la visualización y búsqueda de datos almacenados en dispositivo con linux embebido.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 12 de diciembre de 2017

Vº. Bº. del Tutor:

D. Jesús Enrique Sierra García

Resumen

El objetivo final de este trabajo es realizar una herramienta de graficado, que permita usarse en dispositivos linux con poca capacidad de procesamiento. Esta herramienta ha sido diseñada concretamente para el funcionamiento en dispositivos que actúan como cajas negras en vehículos de guiado automático [31], aunque esta herramienta es portable a otras plataformas siempre que se dispongan los datos de la misma manera.

La herramienta realiza, antes del graficado, una extracción de datos de forma dinámica a través de un XML, donde se marca la disposición en bits de las variables dentro de las tramas que contienen los ficheros de la caja negra.

Esta herramienta gráfica invoca funcionalidades de forma automática de la librería gráfica utilizada, aunque el grueso del trabajo son las funcionalidades propias, entre ellas destacan: la búsqueda de datos, filtraje de datos, auto-diezmado con zoom, ventana temporal, modo reproducción, escalado, umbrales, ajuste automático, y vistas personalizables: en las cuales podremos exportar e importar vistas para su posterior uso con otros archivos/dispositivos.

Para todo esto he desarrollado esta herramienta, cuya utilización está enfocada únicamente a la monitorización de variables, con el objetivo de ser utilizada por técnicos, haciendo uso del navegador web, adaptado para dispositivos móviles o de escritorio.

Descriptores

Gráficas, herramientas de graficado, ChartJS, Javascript, Bootstrap, HTML5, PHP

Abstract

The final objective of this work is to realize a graphing tool that could be used with linux devices with little processing capacity. This tool has been specifically designed to operate in devices that act as black boxes in automatic guided vehicles, although this tool is transferable to other platforms if they are provided with the data and if it is arranged in the same way. The tool performs a data extraction, dynamically, through a XML, where the bit layout of the variables within the files contained in the machine in each frame containing several files of the black box.

This graphical tool invokes features of the graphic library used automatically, although the bulk of the work are its own functionalities, among them we can highlight: data search, data filtering, auto-decimation with zoom, temporary window, playback mode, scaling , thresholds, automatic adjustment, and customizable views: in which we can export and import views for being used later with other files / devices.

I have developed this tool for this aim, whose use is focused only on the monitorization of variables, with the aim of being used by technicians, making use of the web browser, adapted for mobile or desktop devices.

Keywords

Graphs, Graphing Tools, ChartJS, Javascript, Bootstrap, HTML5, PHP

Índice general

Índice de figuras

Índice de cuadros

Introducción

En esta memoria se muestra el desarrollo de una herramienta gráfica que sirve para monitorizar variables, como ya hemos mencionado en la introducción, estará alojada en un servidor web local, (lightppd) dentro del mismo dispositivo.

Dicho servidor es un equivalente a un servidor apache con php y donde la aplicación estará enteramente programada en JavaScript y HTML5, todo esto dentro del datalogger/caja negra.

Únicamente usaremos php para mapear los ficheros disponibles en la caja negra que queramos graficar.

Una vez que tenemos los archivos mapeados, la caja negra que tiene otros servicios corriendo(recordamos que es un dispositivo con linux embebido muy limitado) se encargará únicamente de enviar los archivos necesarios al cliente, entre ellos el contenido de la página y archivos javascript. De esa forma nos aseguramos que la carga de CPU del datalogger es mínima.

Esto es imprescindible, porque al ser un elemento industrial, no nos podemos permitir que se detenga o se congele la máquina por culpa de sobrecarga en la CPU o en memoria, ya que se pararía toda la producción y funcionamiento de todos los elementos que sean controlados o gestionados por ella.

Figura 1.1: Imagen del dispositivo en cuestión. Fuente fabricante: http://www.matrix.es

Es por ello, que la herramienta para realizar todos los cálculos, extracción de datos y funcionalidades está programada en JavaScript, el cual, deja todo el peso de ejecución a la maquina cliente.

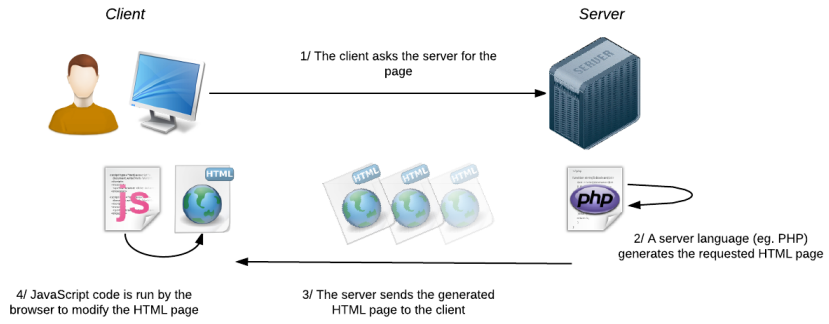

Figura 1.2: Diagrama funcionamiento de un proyecto con JavaScript. Fuente: https://openclassrooms.com

La máquina cliente es la encargada de cargar la biblioteca de variables. En primer lugar, tenemos que parsear el XML maestro, el cual, contiene la disposición en bits de la trama. Éste está codificado en hexadecimal en formato littleEndian.

Una vez extraídas la disposición de las variables gracias al XML, analizaremos todos los ficheros seleccionados y crearemos una biblioteca con esas variables.

Una vez obtenidas dichas variables, junto a todos los datos obtenidos correspondientes a los ficheros, el usuario podrá invocarlas a una gráfica con distintas vistas y opciones.

Esta gráfica ha sido optimizada para que sólo se puedan cargar un máximo de puntos, de manera que vayan de forma fluida en el navegador, por lo tanto, se realiza un diezmado de datos. No se desprecia ningún punto, puesto que la gráfica tiene la funcionalidad, gracias a la ventana temporal, de hacer zoom y calcular de nuevo un diezmado, mostrando hasta ese cierto máximo de puntos fijado.

Aparte de esas funcionalidades, la herramienta tiene otras como puede ser un modo de reproducción, filtrados, búsquedas, cursores, comparación en tabla de varios cursores, modo XY, umbrales, escalados, entre otras, a disposición del usuario; de las que hablaremos de ellas en esta memoria con más profundidad.

Objetivos del proyecto

2.1. Objetivos generales

En esta memoria, se desarrolla una herramienta de graficado, con carácter de herramienta de soporte, para ser empleado por técnicos especializados para monitorear variables. Este proyecto, ha sido aplicado directamente a un caso práctico para monitorear variables de vehículos de guiado autónomo (AGV).

2.2. Objetivos funcionales

Los objetivos funcionales de esta aplicación son:

- El técnico o ingeniero, en adelante usuario, podrá conectarse a través de la red que crea el dispositivo con linux embebido, en adelante caja negra o datalogger, con un móvil, ordenador o tablet al dispositivo, en el que accediendo a una ip predefinida (192.168.5.1) podrá dar uso y disfrute de la aplicación.
- El usuario seleccionará los ficheros donde quiere obtener los datos deseados a monitorear.
- El usuario en todo momento podrá cargar otro XML, y obtener nuevos datos junto a nuevas variables de los ficheros seleccionados.
- El usuario podrá seleccionar el tipo de gráfica que quiere visualizar, donde, podrá elegir en tipo de gráfica temporal o gráfica XY, con distintas opciones únicas dentro de la gráfica.
- El usuario podrá seleccionar la vista de variables, seleccionando una variable y otorgándole ciertos atributos como umbrales o escalados de los datos, así como ciertos atributos dentro de la gráfica, como puede ser el color dentro de la gráfica.

- Los 2 puntos anteriores serán reconocidos como vistas, que el usuario podrá importar y exportar para posibles ficheros, o para graficar únicamente ciertas gráficas.
- Aparte, la herramienta tiene un modo de graficado rápido en el cual únicamente nos dirá que tipos de variables queremos ver, así como los distintos ficheros, y el tipo de gráfica, el cual nos generará una vista por defecto, la cual, podremos modificar o personalizar en todo momento.
- Tanto en la gráfica temporal, como en la XY, tendremos una gráfica maestra, o gráfica temporal, la cual veremos en pequeño, debajo de la gráfica principal. Esta nos permitirá seleccionar periodos de tiempo deseados, así como la evolución de la gráfica en el tiempo a través de la función de reproducción. La cual nos permitirá observar la evolución en el tiempo, con los fragmentos deseados por él.

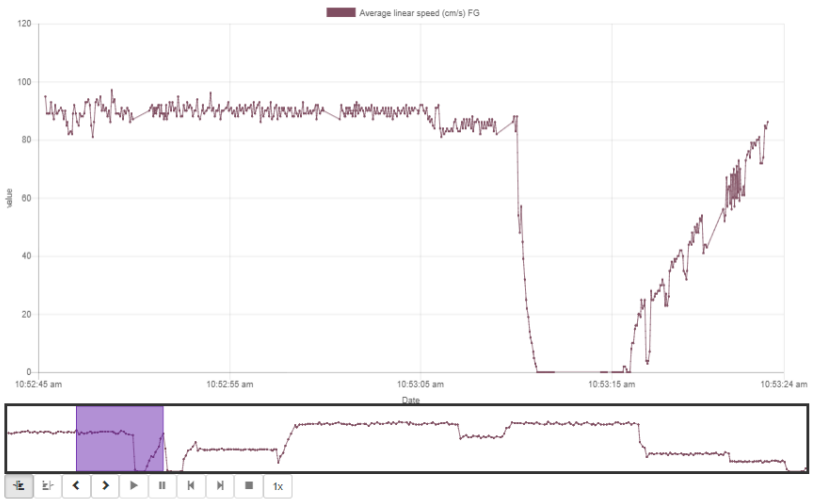

Figura 2.3: Ventana temporal resultante del proyecto

- En todo momento la gráfica principal mostrará un máximo de 700 puntos, y la gráfica maestra o temporal un máximo de 300. Como en la mayoría de casos el número deseado de casos el número de variables es superior a estos números, la herramienta realizará un diezmado. Cada vez que se aplique un zoom, los intervalos de tiempo serán diferentes, por lo tanto, el diezmado también lo será.
- Únicamente en la gráfica temporal, tendremos a nuestra disposición 2 cursores, que podremos situar en la gráfica a nuestro gusto. Cuando se sitúen los 2 dentro de la gráfica, saldrá una tabla comparativa de los valores en el eje Y, capturando los valores de cada variable en cada uno

de los 2 cursores, y comparando los valores en cada instante de tiempo que está situado el cursor.

Figura 2.4: Imagen con dos cursores con su respectiva tabla del proyecto.

- La herramienta gráfica tiene una búsqueda de datos, en la cual, tras realizar una consulta, te colocará un cursor una vez encontrado el primer valor. A partir de ese cursor se realizarán distintas búsquedas con referencia de ese cursor. Esta funcionalidad tiene el añadido que te realizará un zoom automático para que veas que ese punto existe, puesto que es posible que debido al diezmado no esté visible ese punto.
- Además, tendremos un filtraje de datos, en la cual la herramienta solo te graficará los puntos que cumplan la consulta.

2.3. Objetivos tecnológicos

- Los objetivos tecnológicos utilizados en esta aplicación han sido:
- La aplicación tenía que tener la menor carga posible para el Datalogger.
- Utilización de bootstrap[13] para lograr que la aplicación web sea responsive, de tal manera que se pueda utilizar tanto en plataformas de escritorio como web.

Figura 2.5: Diagrama responsive. Fuente: http://johnpolacek.github.io

- Desarrollo de la aplicación con el formato single-page application, de tal manera que ofrecemos desde un único sitio web toda la herramienta de principio a fin[30], simulando una aplicación de escritorio para los usuarios y por ello dando una sensación de aplicación compacta y fluida.
- PHP 5.5 para mapear los ficheros del dispositivo.
- Ejecución enteramente en javascript, para que el dispositivo solo tenga que enviar datos al dispositivo cliente, salvo cuando se realiza el mapeo de ficheros en PHP, ofreciendo por tanto seguridad en el contenido de los archivos y apenas dando carga al dispositivo.
- Página moderna gracias a la utilización de HTML5
- Página dinámica, por la cual puede interactuar el usuario, gracias a la utilización de la liberia de JQUERY.

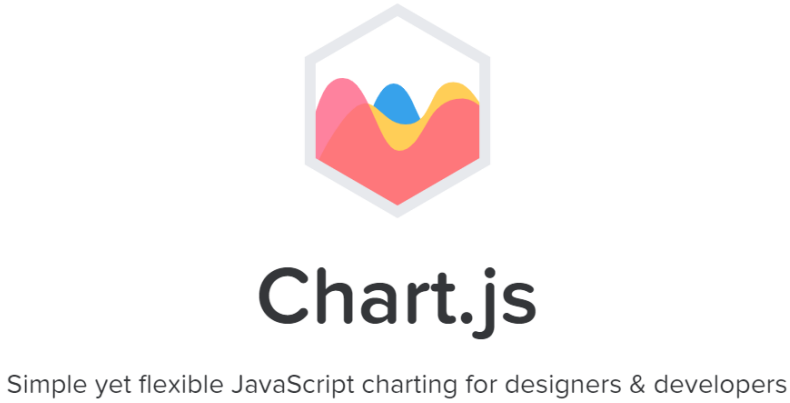

Figura 2.6: Portada de la página oficial chartjs

- Utilización de librería de gráficos (Chartjs) en la cual podamos añadir nuevas funcionalidades a las ya disponibles.

Conceptos teóricos

3.1. Funcionamiento de las cajas negras

Para este proyecto se nos asignó el dispositivo MTX-GTW [5], el cual es un dispositivo con linux embebido que actúa como caja negra en los distintos vehículos de guiado autónomos. Este dispositivo, no solo actúa como caja negra, sino que realiza otras tareas más técnicas como puede ser resolver problemas de tráfico, etc.

Los ficheros con los que tenemos que trabajar en el proyecto, son generados a través de scripts de Python en los cuales almacenan la trama que es recibida a través del conector CAN 2.0B [14] del dispositivo.

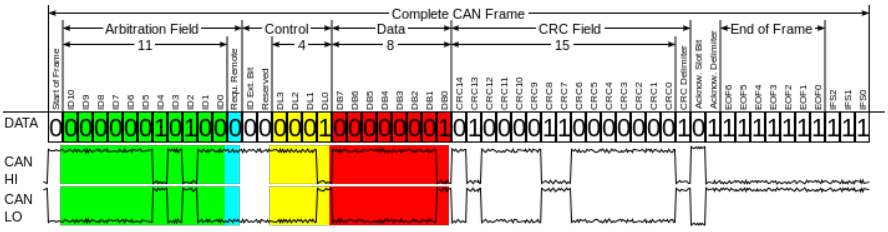

Figura 3.7: Diagrama de un BUS CAN. Fuente: wikimedia.org.

Estas líneas, o tramas tienen el siguiente formato:

30 09:58:04:534923 03C 1B00060004110000

Donde 30 sería el día, 09:58:04:534923 sería la hora hasta en microsegundos, 03C es el identificador del contenedor de la trama de variables, en el cual, tendremos que especificar en el XML como dispondrá dicha trama, y que variables va a contener.

El contenido de la trama se almacena en 1B00060004110000 el cual almacena una o varias variables.

```
30 10:03:28:669199 03E#F3038D39A1380A00
30 10:03:28:683174 040#BE004D0004000000
30 10:03:28:684774 047#B5E6140000000000
30 10:03:28:686314 04C#14080014FFFF7964
30 10:03:28:687805 03D#73B4020004840000
30 10:03:28:711153 03E#FF03D03941381200
30 10:03:28:712760 040#BE00820004000000
30 10:03:28:714295 047#B6E6140000000000
30 10:03:28:715832 04C#16080014FFFF7A64
30 10:03:28:717405 03E#FF038D39F3371600
30 10:03:28:718933 040#BF00870004000000
30 10:03:28:731177 047#B5E6140000000000
30 10:03:28:734554 04C#14080013FFFF7B64
30 10:03:28:736156 03E#F503AF39D0390800
30 10:03:28:737698 040#C000F5FF04000000
30 10:03:28:739522 047#B6E6140000000000
30 10:03:28:751734 04C#14080014FFFF7C64
30 10:03:28:753307 03E#F503D039C2380C00
30 10:03:28:754853 040#C000580004000000
30 10:03:28:758137 04C#14000012FFFF7D64
30 10:03:28:759910 03E#FB03D039D9380E00
30 10:03:29:772171 040#C100510004000000
30 10:03:29:773696 04C#14000014FFFF7E64
30 10:03:29:775215 03E#FB03D03975381200
30 10:03:29:776730 040#C200730004000000
30 10:03:29:778432 04C#14000013FFFF7F64
30 10:03:29:782467 03E#FB03133A63390E00
30 10:03:29:784063 040#C2003A0004000000
30 10:03:29:785618 047#B7E6140000000000
30 10:03:29:787160 04C#14080013FFFF8064
30 10:03:29:788683 03E#FB03D03977381200
30 10:03:29:801296 040#C300730004000000
```

Figura 3.8: Ejemplo de una pequeña cantidad de tramas que analiza la herramienta del proyecto.

Hay que especificar en esta memoria, que cada variable es única para contenedor, y que tendrán por tanto identificadores únicos. A la hora de escoger lenguaje y plataforma del dispositivo, la manera óptima de realizar este proyecto hubiera sido tratar con una base de datos este tipo de variables y que corriera todo el servidor en el dispositivo, pero esto no fue posible por 2 motivos:

- No se podían instalar programas nuevos puesto que era un sistema Linux muy básico, y estos programas tenían que venir instalados por parte del fabricante, el cual mediante cadenas de compilación se tenía que adaptar a ese sistema operativo concreto.
- Al ser un elemento de un proceso industrial, con capacidades de procesamiento muy limitadas, con 458 Mhz, y apenas 128 MB de RAM, con otros procesos activos, como habíamos mencionado, corriendo en segundo plano, la opción de correr en la parte del servidor no era válida, puesto que era más que probable que el dispositivo se congelase, y el procesamiento de extracción de variables sería extremadamente lento.

3.2. Pasos del proyecto

En el ciclo de vida del proyecto, se intento dotar la menor carga posible al datalogger. Para ello, el grueso de las operaciones de cálculo, como sería calcular todas las bibliotecas y operaciones de filtros y búsqueda de datos, se encuentra en la parte del cliente.

Dispositivo	Datalogger	Navegador web cliente
Almacenamiento en ficheros de tramas CAN.	X	
Envío de contenido de página.	X	
Mapeo de los ficheros del dispositivo. Servidor PHP.	X	
Envio de los ficheros del dispositivo.	X	
Aplicación del XML a las tramas de los ficheros.		X
Almacenamiento de las tramas en la biblioteca.		X
Parseo del XML en javascript.		X
Utilización y graficado de dichas bibliotecas.		X
Filtros y búsquedas.		X
Exportación e importación de vistas.		X

Cuadro 3.1: Dispositivo en el que son ejecutadas cada proceso de la vida del proyecto

3.3. Conceptos de lenguajes

3.3.1. XML

Figura 3.9: Logo XML. Fuente: Freepik.es.

Es un lenguaje cuyo objetivo es diseñar lenguajes de marcado. Este fue creado para poder ofrecer la información de la forma más estructurada, abstracta posible, con la opción de poder ser reutilizable.[33]

Es por eso que ha sido utilizado en nuestro proyecto para especificar la disposición de las tramas.

Escogí XML frente a otros lenguajes para establecer la configuración de las variables puesto que es mucho más visible y estructurado de cara al usuario a la hora de especificar una configuración, ya que este XML tiene el objetivo de ser continuamente adaptado y modificado.

En nuestra estructura sigue el siguiente formato, donde puede haber distintas tramas por configuración, y distintas viables por trama:

```
<configuracionTrama>
    <trama id="{contenedor variable}">
        <var id="{nombreVariable}" >
            <tipo>{formato variable}</tipo>
            <startBit>{bit comienzo}</startBit>
            <endBit>{bit final}</endBit>
        </var>
    </trama>
</configuracionTrama>
```

Figura 3.10: XML con la estructura de la disposición de tramas

Como se aprecia, el lenguaje es muy fácil de entender y es por eso que fue seleccionado como fichero de configuración a la hora de la lectura dinámica de variables.

Se estudió añadir XSD [12](documento esquema), que es un convenio fijado para la estructura de un XML. Se decidió al final no emplearlo puesto que en cierto modo limitaba la libertad que XML nos daba, y no era necesario para el funcionamiento o marcar cierto orden en la estructura, puesto el XML ya de por si, considerábamos que estaba suficientemente estructurado.

3.3.2. JSON

Figura 3.11: Logo JSON. Fuente: https://qph.ec.quoracdn.net/.

JSON es un formato de lenguaje independiente, el cual trabaja como notación de objetos en JavaScript.

Fue seleccionado para importar y exportar vistas, puesto que estando trabajando con JavaScript es perfecto para importar y exportar objetos.

Como JSON es formato de texto ligero, permite pequeñas modificaciones en dichas vistas siempre y cuando sean compatibles con las herramientas. [26]

Por lo tanto, fue seleccionado, por encima de la alternativa de XML en esas tareas concretas.

3.3.3. CSS

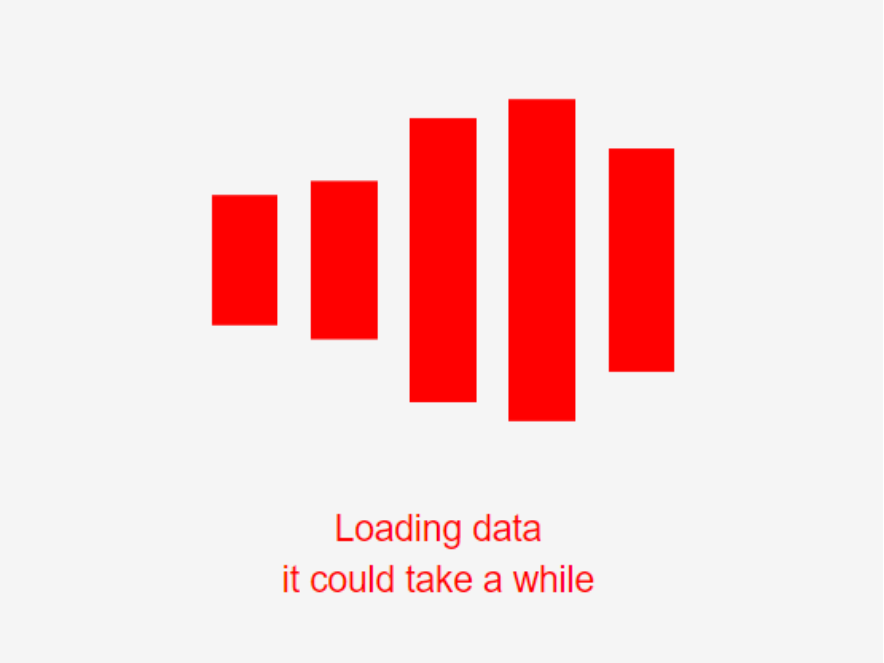

Figura 3.12: Animación con CSS. Gracias a CSS podemos realizar imagenes como esta.

Conocido en español como hoja de estilos, es un lenguaje utilizado para definir los estilos y propiedades respecto a la estructuración de los distintos elementos que se encuentran en una página. [22]

Surge con la idea de separar la estructuración de dichos elementos fuera del HTML.

La versión aplicada en este proyecto es CSS3.

3.3.4. PHP

PHP es un lenguaje de programación libre, el cual está publicado con una licencia libre similar a la GNU, llamada PHP, con algunas restricciones. Este lenguaje tiene el objetivo de ser empleado en el servidor (parte del servidor), siendo de los primeros lenguajes (fue creado en el 1995 por Rasmus Lerdorf)

que se podían incorporar directamente contenido dinámico a través de un documento HTML.[6]

El código que contiene, es procesado por el servidor, gracias al módulo de procesador de PHP, donde este genera el contenido dinámico antes mencionado. Además, PHP provee al programador de una interfaz con línea de comandos, para ayudar a este en posibles tareas como pueden ser aplicaciones gráficas.

Figura 3.13: Logo PHP 5.5. Es la versión utilizada en este proyecto. Fuente: https://community.dynatrace.com

La potencia de PHP reside en que puede usado en casi todos los sistemas operativos y plataformas, lo cual le convierte en un lenguaje universal y gratuito.

PHP es criticado normalmente por ciertos aspectos en la actualidad. Entre estos motivos son:

- La lentitud de un script de PHP respecto a un lenguaje de bajo nivel, cuyo efecto puede ser reducido gracias a usar en archivos y en memoria técnicas de cache.
- En versiones anteriores a la 7, las cuales en la actualidad (2017), apenas nadie usa, las variables ofrecen un tipado flojo, esto conlleva que a la hora de utilizar la mayoría de los IDEs no nos puedan ofrecer la asistencia común que nos ofrecen otros IDES.

Es un lenguaje difícil de ofuscar. Esto conlleva que cualquiera puede entender nuestro código, aunque nosotros no queramos, con lo que esto conlleva.

Nosotros, como ya hemos mencionado en la memoria de este proyecto, nada más usamos PHP para que mapee 2 directorios, que es donde se encuentran los ficheros. En el cual, con una llamada síncrona, pasamos el directorio listado.

3.3.5. Javascript

Conocido por sus siglas JS, es un lenguaje de programación interpretado, débilmente tipado y orientado a objetos. Está basado en el tipificado ECMASCRIPT como dialecto Es utilizado normalmente en el lado del cliente, (no tenemos en cuenta la versión de JavaScript para servidor, SSJS), el cual es interpretado por los distintos navegadores web, concediendo la posibilidad de crear web dinámicas.[24]

Actualmente todos los navegadores modernos (desde el 2012) soportan la versión de JavaScript ECMASCRIPT (5.1)[17].

Figura 3.14: Evolución JS/ECMASCRIPT. Fuente: http://adrianmejia.com

La sintaxis de este lenguaje es una mezcla de lenguajes, el cual es bastante similar a C, pero utiliza ciertos aspectos de Java, como pueden ser nombres y convenciones.

JavaScript suele ser normalmente confundido por Java, pero no tienen nada que ver como lenguajes, ya que son completamente diferentes.

JavaScript interactúa con el DOM de la página (document object model), en el cual están todos los objetos de la página, que como hemos dicho, con los que puede interactuando, aunque este es uno de los distintos usos utilizados.

Uno de los principales inconvenientes de JavaScript, es que programadores usan este lenguaje de forma inadecuada y con fines ilícitos, con relativa facilidad, por lo que es una de las vías de insertar código malicioso vía web.

Uno de los principales debates entre programadores [8], es que hay algunos no consideran JavaScript como un lenguaje orientado a objetos. Yo considero este lenguaje como un lenguaje orientado a objetos, pero técnicamente y estrictamente no lo es, porque para que un lenguaje sea orientado a objetos tiene que cumplir las siguientes condiciones:

- Tiene que soportar el polimorfismo, que si que lo cumple. Puesto que la mayoría de los lenguajes dinámicos lo hacen.

- Tiene que soportar la encapsulación, lo cual también lo cumple, puesto que una de las principales características de JavaScript es la facilidad de encapsulación que este ofrece.
- Sin embargo, el último atributo es la herencia, la cual técnicamente no la soporta, únicamente permite la herencia a través de prototipos, pero técnicamente esto está a expensas de la encapsulación. De hecho, en nuestro proyecto no hemos utilizado la herencia por esto mismo.

Aun así, JavaScript sigue siendo para mí un lenguaje orientado a objetos, puesto que podemos implementar un OOD (Diseño orientado a objetos) en JavaScript.

3.3.6. HTML5

Figura 3.15: Logo HTML5. Fuente:http://oleblogs.com

Es la quinta versión, y la más moderna versión utilizada en la actualidad, por el lenguaje de marcado HTML, este se utiliza para estructurar el contenido de las páginas web. [10]

Esta versión está regulada por W3C [32], el cual es un consorcio internacional cuyo objetivo es asegurar el crecimiento la web a largo plazo.

La sintaxis de este lenguaje consiste en etiquetar una serie de elementos, en forma de contenedor, con una serie de etiquetas, que pueden contener o no contenido propio, por el cual, el navegador web interpretará y mostrará al usuario.[23]

HTML5 además incorpora códecs para mostrar contenidos multimedia como videos, de forma que se ha ido el mermando el uso de flash player en la web, el cual contenía muchas vulnerabilidades y sustituyendo por reproducción nativa HTML5. También destaca la incorporación de etiquetas para manejar muchos datos, ciertas mejoras en los formularios para evitar el uso de JavaScript, visores para fórmulas matemáticas (MATHML) , funcionalidades para arrastrar todo tipo de objetos como imágenes y visores de gráficos vectoriales (SVG).

Técnicas y herramientas

4.1. Técnicas de desarrollo

4.1.1. Metodología ágil

Para el desarrollo de este proyecto se ha resuelto mediante metodología ágil.

Como resultado de los resultados de las antiguas metodologías, surgen nuevas soluciones en el tiempo, con las cuales se consigue un aumento en el desarrollo software basándose en dos principios:[16]

- Postergar las decisiones.
- Planificación adaptativa, es la planificación que otorga a los miembros del equipo de desarrollo poder, para que puedan estar preparados para posibles modificaciones en el transcurso del proyecto, incluso la toma de decisiones mientras transcurre este último.

Esto conlleva que al cliente se le vayan entregando versiones funcionales del proyecto de software, cada vez más completas, desde una versión básica hasta conseguir el resultado deseado por el cliente.

Estas nuevas versiones, cada vez más funcionales pueden ser cambios o modificaciones, que el equipo de desarrollo tendrá que hacer frente, de tal manera, que sin aumentar el coste se desarrolle un software flexible para tener la capacidad de aceptar.

En un caso práctico en nuestro proyecto, en un principio se optó por utilizar una librería totalmente distinta, que no tenía nada que ver con la entrega del proyecto. Gracias que en un principio los datos estuvieron desarrollados esperando posibles cambios, el código de extracción de datos apenas sufrió modificaciones.

Por tanto, podríamos resumir un proceso de metodología ágil, a aquel que cumpla las siguientes características:

- Existan numerosas entregas de software funcional
- Comunicación dentro del equipo y con el cliente durante todo el proyecto
- Continuidad y motivación en el tiempo de desarrollo
- Estado del proyecto se mide por la funcionalidad de la aplicación deseada por el cliente.

4.1.2. SCRUM

Figura 4.16: Logo Scrum. Fuente: https://www.codercamps.com.

Dentro de las distintas metodologías ágiles hemos escogido la metodología scrum, que nos ha permitido dirigir y administrar el desarrollo del software.

Como característica de metodología ágil, hemos ido realizado en el desarrollo del proyecto distintas versiones incrementales, en las cuales hemos añadido nuevas funcionalidades y cambiando/arreglando otras[28].

En primer lugar, se llevó a cabo una definición completa de requisitos con el cliente, en tal caso, el tutor del proyecto.

Una vez establecido los requisitos tutor y alumno acordamos distintas fases en la ejecución del proyecto, marcando distintos tiempos en la ejecución de estos.

Semanalmente, iba informando del progreso del proyecto con el cliente, en este caso con el tutor, y le mostraba las distintas funcionalidades que había aplicado en cada iteración, en muchos casos, insatisfactorias, por lo que se procedía en la reunión a recoger las posibles modificaciones en dichas versiones, y modificar nuestras fases para adaptar estas modificaciones.

La inspección final de la revisión era cuando el cliente daba el visto bueno a dichas iteraciones, dándome pie a trabajar en otros nuevos.

Uno de los pasos de scrum, es que se cuenta con un equipo con distintos roles.

Al realizar este proyecto una sola persona, el reparto no fue equitativo y resultó que adquirí más de un rol.

Al tratarse de un proyecto realizado por una sola persona, se trató al tutor como cliente (product owner), y yo adquirí las tareas de scrum máster y de equipo de desarrollo: anotando y preparando nuevas iteraciones, así como encargándome de su desarrollo integral.

Esta metodología ágil, denominada Scrum, como podéis ver, no puede considerarse rígida en su planteamiento, y seguramente otros alumnos hayan realizado esta metodología en sus proyectos con diferentes variantes, pero cumpliendo siempre con los principios y adaptando las necesidades del proyecto al equipo de personas que participan en la metodología SCRUM.

4.2. Herramientas de documentación

4.2.1. LaTeX

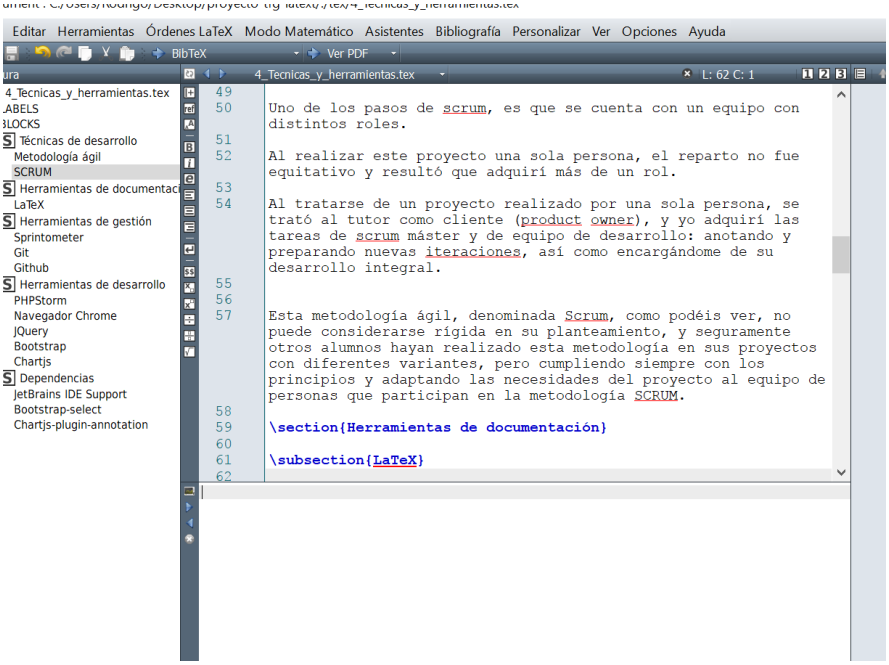

Figura 4.17: Captura del editor TexMaker que trabaja con Latex.

Latex es un editor de texto que permite ser usado para realizar artículos científicos por su alta calidad tipográfica a partir de macros. [27] En el proyecto hemos seleccionado esta opción pues el resultado era definitivamente mejor que en OpenOffice, aunque presentó alguna dificultad en su puesta en marcha.

Para este trabajo se ha utilizado el editor de latex TexMaker [9],que es gratuito bajo la licencia GPL.

Página oficial editor Latex: http://www.xm1math.net/texmaker/

4.3. Herramientas de gestión

4.3.1. Sprintometer

Figura 4.18: Captura de la página oficial de Sprintometer.

Sprintometer es una herramienta de seguimiento de metodología ágil para proyectos Scrum y extreme programming. Esta herramienta en principio fue creada para trabajar en proyectos ágiles con sus propios propósitos y ahora es una herramienta freeware.[7]

Hemos usado esta herramienta para registrar los sprints, a su vez divididos en tareas para después comparar los tiempos estimados con los reales.

Página oficial: http://sprintometer.com/

4.3.2. Git

Figura 4.19: Logo de Git.Fuente: git-for-windows.github.io

Hemos utilizado git, como software de control de versiones[19]. Este nos ha permitido controlar las distintas versiones de nuestro proyecto. Su uso es gratuito y tiene una licencia GNU GPL V2.

4.3.3. Github

Se ha escogido esta plataforma de desarrollo colaborativo, ya que este proyecto es open source.

Su utilidad es para almacenar en la nube, gracias a git, el control de versiones de un producto[20].

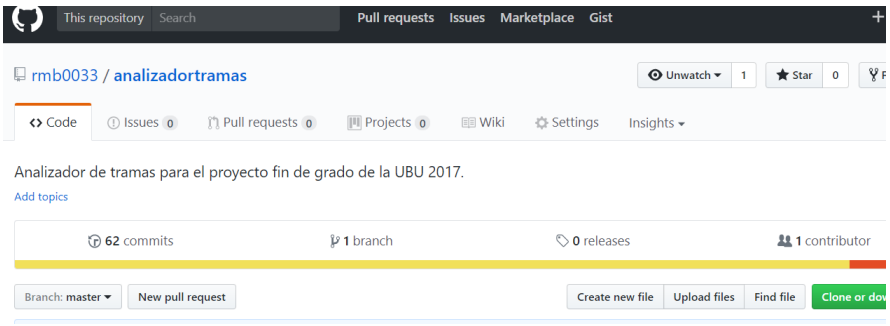

Figura 4.20: Captura del estado del proyecto en github.

Github no es siempre es gratuito, y tiene distintos tipos de usuarios. Nosotros usamos el modelo gratuito el cual nos obliga a que nuestros proyectos sean públicos. Página: http://github.com

4.4. Herramientas de desarrollo

4.4.1. PHPStorm

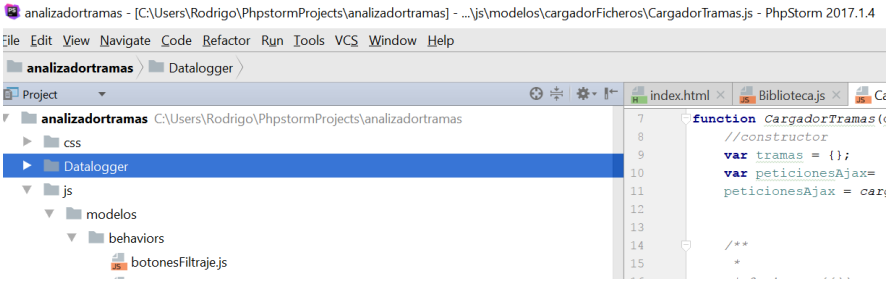

Figura 4.21: Captura del proyecto en phpStorm.

PHPStorm es un entorno de desarrollo para PHP, cuyo propietario es IntelliJ.[3] He utilizado enteramente este IDE para el desarrollo del proyecto, y como IDE principal. Como casi todos los productos de IntelliJ, no es un producto gratuito, pero jetbrains ofrece a los estudiantes y profesores licencias gratuitas. Este IDE es de los más valorados de la comunidad y lo he escogido por las siguientes características:

- Necesitaba un IDE tanto para PHP como para JavaScript, y PHPStorm me lo ofrecia. (WebStorm[4] es un IDE especifico de JavaScript

del mismo fabricante, pero phpstorm nos ofrecía lo mismo respecto a la funcionalidad).

- Tiene un potente control de versiones, muy intuitivo y fácil de usar respecto a otros IDEs que he trabajado (Eclipse).
- El IDE es muy robusto. En el desarrollo de todo el proyecto, no he visto ningún bug notable en el IDE.

https://www.jetbrains.com/phpstorm/

4.4.2. Navegador Chrome

El navegador Chrome es el navegador líder en la actualidad, con la mayor cuota de mercado. Es por eso que fue nuestra decisión depurar en él frente a otras alternativas.

El navegador Chrome nos permite realizar estas tareas como desarrolladores: -Utilizar la consola web. -Depurador web y javascript. -Cambiar la resolución del dispositivo, incluso cargar resoluciones predeterminadas. -Inspector de páginas. -Editor CSS -Analizador de rendimiento -Analizador de memoria. -Analizador de tráfico de red. Link: https://www.google.es/chrome/browser/desktop/

4.4.3. JQuery

JQuery es una biblioteca JavaScript, con la que gracias a ella, nos permite interactuar con objetos del DOM de manera sencilla, con ciertas funcionalidades extra, la cual la convierte en la biblioteca JavaScript más usada en la actualidad. Tiene una licencia MIT por lo que podemos darle uso gratuito reconociendo al autor.

En el proyecto la utilizamos para modificación de css, efectos, animaciones y utilización de elementos del DOM.[25]

Link: https://jquery.com/

4.4.4. Bootstrap

Utilizamos bootstrap, una biblioteca de herramientas para frontend, por el gran contenido que nos aporta a la página, ya que abstrae al programador en ciertos elementos predefinidos en la biblioteca, entre las que destacan hojas de estilos, que a su vez están combinados con código en javascript, lo que nos aporta una gran cantidad de recursos para diseño web. Una de las ventajas de usar bootstrap, es convertir nuestra página en una página responsive.

Link: http://getbootstrap.com/

4.4.5. Chartjs

Chartjs es la librería que hemos seleccionado para realizar nuestro proyecto. Tiene una licencia MIT, aspecto importante que vamos a explicar más adelante en nuestra memoria y anexos.[15]

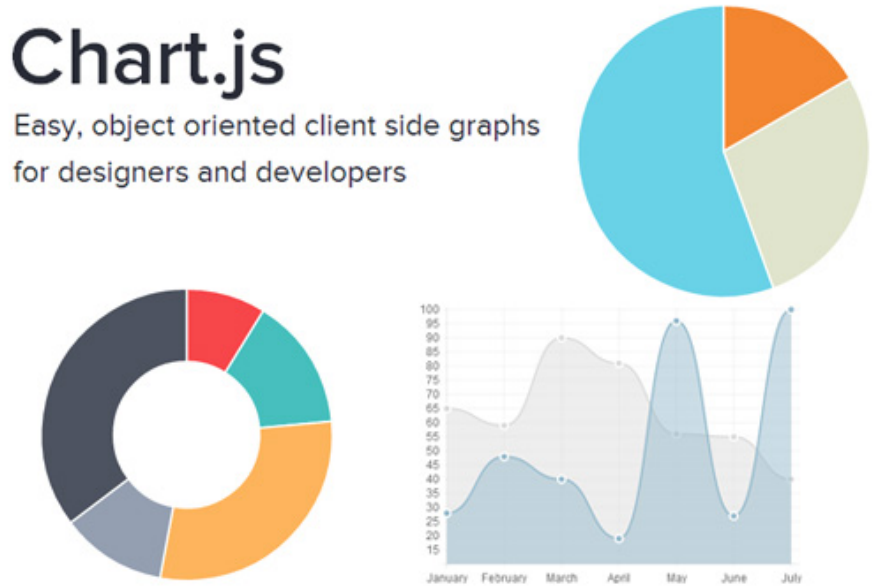

Figura 4.22: Banner de ChartJS. Fuente http://muchocodigo.com

La selección de la librería para gráficos es clave en este proyecto, ya que se basa en una herramienta de graficado. Es responsive, utiliza el canvas de HTML5, está bien documentada, es open source,tiene multitud de personalizaciones y tiene un gran soporte de la comunidad. Link: http://www.chartjs.org/

4.5. Dependencias

4.5.1. JetBrains IDE Support

Utilizamos esta extensión de Chrome para depurar en PHPStorm mientras tenemos una página abierto en Chrome. Permite depurar en vivo la página mientras operamos con ella.

Link: https://chrome.google.com/webstore/detail/jetbrains-ide-support/hmhgeddbohgjknpmjagko

4.5.2. Bootstrap-select

Bootstrap-select es un plugin de Jquery que utiliza bootstrap, y lo hemos empleado para todas las cajas de selección (todas las listas con texto) de

nuestra herramienta de graficado.

4.5.3. Chartjs-plugin-annotation

Utilizamos este plugin de ChartJS, ya que permite escribir líneas y cajas dentro del canvas de la gráfica, ya que de forma nativa no te deja. Aun así, estas cajas y líneas son dibujadas a partir de datos dentro de la gráfica, por lo que hemos tenido que trabajar en una solución para que pudiera dibujar a nuestro gusto los cursores por ejemplo, desconociendo el punto exacto una vez que no es contemplada en este plugin.

Link: https://github.com/chartjs/chartjs-plugin-annotation

Aspectos relevantes del desarrollo del proyecto

5.1. Conceptos técnicos en las variables. Formato.

Como habíamos comentado, las variables se almacenaban en las tramas que contenía el fichero. Cada línea del fichero correspondía a una trama distinta, y las tramas tenían configuraciones distintas dependiendo del contenedor de la trama/variable.[18] Por tanto, vemos en un ejemplo, cuáles son los elementos que forman parte:

30 09:58:04:534923 03C 1B00060004110000

- 03C es el contenedor de la variable.
- 1B00060004110000, es el contenido en hexadecimal y en littleEndian de los datos.

A continuación, vamos a especificar como se realiza la extracción de variables entre los distintos tipos.

LittleEndian, "de comienzo por el extremo pequeño", es una adopción del formato que se almacenan los datos con más de un Byte.

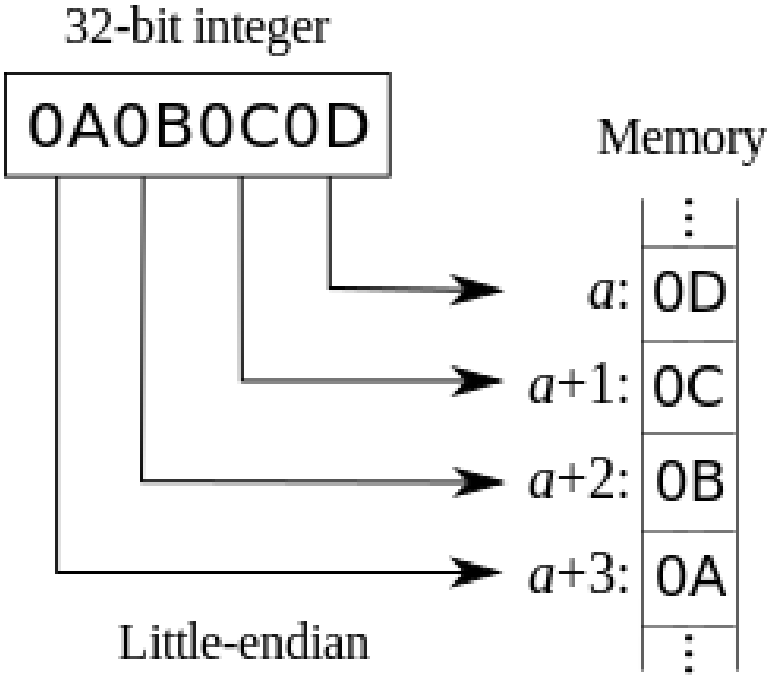

Figura 5.23: Ejemplo disposición LittleEndian. Fuente https://es.wikipedia.org

Este sistema hace más efectivo el acceso a datos, efectuándose de manera incremental de menos relevante a más relevante, dando más importancia a como acaban las cosas más que como empiezan.

Según el fabricante los datos se guardan en LittleEndian o BigEndian, aunque estos 2 no son los únicos formatos.

En el caso de este proyecto nos basaremos únicamente en LittleEndian.

En este ejemplo si tendríamos un entero de unsigned de 32 bits (cogeriamos los primeros 32 bits) se distrubiría de la siguiente forma.

- 1B000600 : Formato hexadecimal.
- 00, 06, 00, 1B: En littleEndian.

Una vez que tenemos los datos en littleEndian, procederiamos a traducir a decimal este valor, el cual puede ser traducido directamente gracias a javascript.

- En binario: 0000 0000 0000 0110 0000 0000 0001 1011
- En decimal: 393243

Un caso más complejo sería si en vez de unsigned, fuese signed.

Entonces tenemos que tener en cuenta el complemento a 2 en binario de dato antes de pasarlo a decimal.

5.2. Fallos del datalogger

Realizando este proyecto, nos percatamos que los datos dentro del datalogger a veces fallaban. Ya que algunas gráficas como resultados daban picos anómalos que no tenían sentido. Estos fueron los datos que nos hicieron percatarnos de esto: 30 09:58:59:999999 30 09:58:50:002123 Lo que ocurría es que el segundero, minutero y el contador de las horas del archivo generador de tramas no estaban incrementando bien, ya que estaba varios relojes del sistema, y fallaba cuando tenía que incrementar, sucediendo solo en los casos en los que los microsegundos tendían a 0 (Observamos distintos casos). Por lo tanto, tuvimos que realizar una función para reparar esos datos y que no se perdiesen esas tramas, puesto que algún dato podía ser crítico. (Despreciamos los errores que surgían justo en el cambio de día, al tratarse de un error muy puntual)

5.3. Analizador del XML en JavaScript

Para este proyecto, tuvimos que implementar el analizador de XML,de JavaScript, con el XML DOM. La tarea no fue sencilla ya que el analizador recogía basura, del DOM, en las raíces de cada nodo, y tuvimos que rechazarla. De hecho, nuestro XML, no acepta comentarios, porque suponía una dificultad extra al proyecto, aunque se marcará como posible mejora

La sintaxis era la siguiente [11] Para invocar al analizador:

```
parser =new DOMParser();

xmlDoc = parser.parseFromString(text,"text/xml");

Y para recorrer sus nodos:

x = xmlDoc.getElementsByTagName("Nodo");

for(i =0; i < x.length; i++) {
    txt += x[i].childNodes[0].nodeValue+"<br>";
}
```

Figura 5.24: Sintáxis del analizador XML en javascript

Decidimos guardar la estructura en un objeto una vez parseado el objeto, para dar flexibilidad al código.

5.4. Complejidad extracción biblioteca

Una de las decisiones que tomamos fue extraer todos los datos en la biblioteca, en vez de ir cargando una a una, las variables según lo pidieran los usuarios. Esta decisión tiene sus ventajas e inconvenientes: Ventajas: Una vez realizada la primera carga, el usuario podrá cargar distintas variables, sin que tenga que realizar ninguna nueva carga, de forma que irá mucho más fluido. Inconvenientes: Mayor uso de memoria. Se realizará una carga completa de todas las variables de los ficheros, por lo que la primera carga durará más

Complejidad algorítmica de la creación de la biblioteca:

- Recorrer ficheros seleccionados (numero ficheros seleccionados)
- Recorrer cada trama de ese fichero (numero ficheros seleccionados * lineas fichero).
- Utilizar configuración para extracción de esa trama en concreto (numero ficheros seleccionados * lineas fichero * distintas configuraciones XML)
- Recorrer la cadena de caracteres para sacar los datos de dichas variables (numero ficheros seleccionados * lineas fichero * distintas configuraciones XML * numero de caracteres trama)

5.5. Conector gráfica-biblioteca

Una vez obtenidos todos los datos, teníamos que dejar las colecciones de forma que las librerías de gráficos pudieran trabajar con esos datos.

Para conectar estos datos tenemos en cuenta los 2 tipos de gráfica:

- En la gráfica temporal, creamos un diccionario como clave: la hora, y valor: el valor numérico de dicha variable, junto la hora otra vez. Es necesario tener un campo adicional para la hora, pues creamos un diccionario para ordenar los valores por fecha (a la librería los datos tenian que ir ordenados). Una vez ordenados estos valores, la clave que recorremos no la reconoce como un objeto tipo fecha, sino como un string, por lo que la herramienta gráfica no reconoce.
- En la gráfica XY: Como en el caso anterior tenemos que ordenar los datos por su fecha, tanto para el valor x como para el valor y. Una vez que ordenamos las fechas, tenemos que sincronizar los 2 valores (el valor x, y el valor y). La forma de sincronización es la siguiente: Si tenemos unos instantes de tiempo para la variable del eje x 1,3,5,6,10 Si tenemos unos instantes de tiempos para la variable del eje y 4,5,6,8 La sincronización se realizará cuando se dispongan ambas variables en el instante de tiempo, en este caso, en el segundo 4 tendremos las variables de x del segundo 3 (hasta el segundo 5 no cambia de valor, por lo tanto, mantiene el valor del segundo 3). Por cada instante de tiempo del valor x, y una vez sincronizadas: tendremos un nuevo valor para la gráfica X-Y. .

5.6. Diezmados

Para un correcto funcionamiento del navegador, los puntos totales mostrados en la gráfica tienen que estar acotados en un máximo, es decir, el navegador no puede ir fluido graficando un número elevado de puntos en un canvas.

Por ello marcamos un máximo de puntos, en cada gráfica.

La característica especial de nuestra herramienta, es que siempre vamos a trabajar con los datos en bruto, por lo que, si se realiza un zoom o una búsqueda, los datos mostrados se recalcularán con diezmados por lo que, si en un principio tenemos 10000 datos entre los segundos 1 y 10, con 1000 datos por segundos. En principio haríamos un diezmado de los 10000 datos, es decir, graficaríamos 1 punto por cada (10000/ puntos máximos diezmado).

Si haríamos un zoom entre los segundos 3, 4, realizaríamos un diezmado de nuevo entre esos 1000 datos, (1 punto por cada (1000/ puntos máximos diezmado)) por lo que es posible que se muestren nuevos picos/puntos que a priori no podíamos ver, ya que habían sido diezmados.

Los diezmados tienen en cuenta las demás variables, y son repartidos equitativamente, de tal manera que siempre se respetará el máximo de puntos graficados.

5.7. Gráfica temporal/maestra

Debajo de cada gráfica temporal, hemos realizado una gráfica maestra, que será utilizada como mapa conceptual con un diezmado más elevado que la superior. En esta gráfica hemos añadido distintas funcionalidades, todas ellas, funcionalidades que no existían en la librería de gráficos utilizada:

- Un cuadro de zoom, en la cual, obtendremos información de que datos coger en la gráfica principal y que diezmado aplicar.
- Distintas operaciones con el cuadro de zoom.
- Simulación de eventos de click, para obtener el punto más cercano dentro de la gráfica y poder utilizarla por ejemplo para marcar límites del zoom.
- Una reproducción, gracias a jQuery, y forzando el sincronismo, observamos la animación de cómo se mueve el cuadro de zoom, pudiendo incrementar/aminorar la reproducción.

5.8. Búsquedas y filtros

Para calcular las búsquedas y filtros, hemos creado una gramática gracias a JavaScript que sigue la siguiente sintaxis. Tokens:

- Variable=nombre de la variable
- Valor = valor número que queremos comparar
- Comparador $(==, =, !=, >, >, >=, <=)$
- Conector (AND, and, OR, or)

Sintaxis:

- Variable comparador valor [conector variable comparador valor]$*$

(Entre corchetes [], opcional, no admite paréntesis)

Esta sintaxis lo que hace es obtener los intervalos de los tiempos en los que se cumplen, de la forma más óptima posible de tal manera que estos intervalos son aprovechados para los filtros o búsquedas.

5.9. Idioma

El idioma escogido para la interfaz de usuario ha sido el inglés, puesto que es el idioma más internacional del mundo. El código del proyecto está en español, puesto que es el lenguaje nativo del programador. Se estudió aplicar internacionalización de la página web, pero se descartó, en primer lugar, porque con un conocimiento básico del inglés la herramienta se domina perfectamente, y entendemos que la gente que puede usar esta herramienta tiene conocimientos del idioma, y, en segundo lugar, porque requería mucho trabajo y se escapaba dentro de los Springs del proyecto.

5.10. Patrón decorador

Hemos utilizado este patrón de diseño para la interfaz gráfica de las vistas (opciones de gráfica, opciones variables) de tal modo que implementábamos a un modal funcionalidades de forma dinámica (objeto del DOM), de tal manera que reutilizábamos el objeto, extendiendo o quitando funcionalidades. [29]

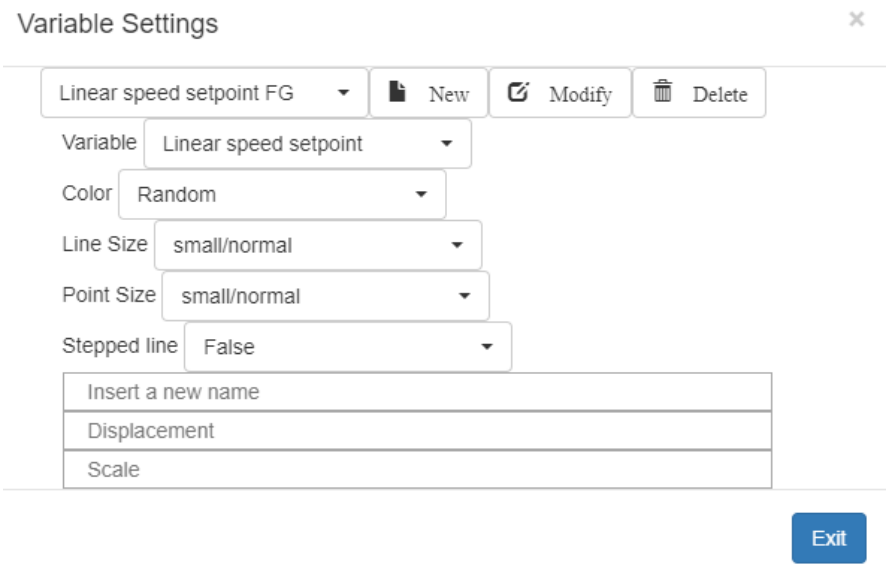

Figura 5.25: Ventana gráfica en la que hemos utilizado este patrón.

5.11. Single-page application

Al tratarse de una herramienta de trabajo, hemos considerado que la mejor opción respecto al diseño era el formato single-page application. Esto lo hemos conseguido a través de inyecciones de HTML mediante JavaScript, y

utilización de spinners para marcar las transacciones entre ventanas, dando la sensación que más de una página web estamos trabajando en una herramienta. Aunque teníamos la opción de trabajar con estados gracias la API de HTML5, permitiendo navegabilidad en páginas lógicas.

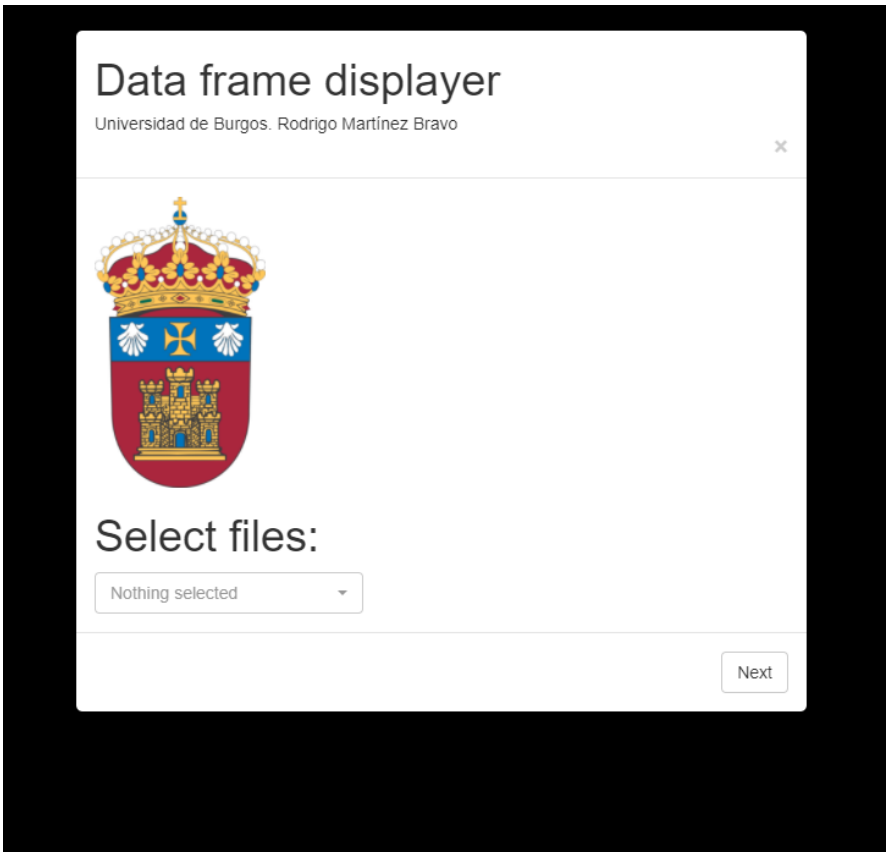

Figura 5.26: Página de inicio SPA.

Como se observa el la imagen, de principio a fin la página no realiza ninguna transacción entre ninguna página. Como contra a este método, podemos decir, que el código se ensuciaba con estas inyecciones de html mediante JavaScript, pero el resultado de la parte funcional es satisfactorio. Respecto al rendimiento, también es la mejor opción, pues la transferencia de ficheros es bastante lenta desde el datalogger , y teniendo este tipo de diseño conseguimos que una vez cargada, la página vaya fluida para el usuario de principio a fin.

5.12. Comparativa herramienta de graficado Chart.js con funcionalidad final del proyecto

Funcionalidades de Chart.js con el plugin annotations

- Actualizar gráfica en vivo
- Dibujar líneas y cuadrados en la gráfica conociendo el tiempo
- Graficar en el tiempo con conjuntos ordenados en el tiempo.
- Modificar aspectos de la gráfica (color, tamaño, etc)
- Graficar coordenadas X, Y.
- Desdibujar a tiempo real variables mediante la leyenda.

Funcionalidades del proyecto realizado.

- Actualizar gráfica en vivo
- Dibujar líneas y cuadrados en la gráfica conociendo el tiempo
- Graficar en el tiempo con conjuntos ordenados en el tiempo.
- Modificar aspectos de la gráfica (color, tamaño, etc)
- Graficar coordenadas X, Y
- Desdibujar a tiempo real variables mediante la leyenda.
- Ventana temporal/ maestra de la dibujada.
- Zoom horizontal visualizando un mapa conceptual del total.
- Reproducción de la gráfica.
- Dibujar tiempos temporales incluso cuando están desordenados.
- Dibujar cursores haciendo click en la ventana de la gráfica
- Calcular puntos reales de la gráfica que corta el cursor.
- Búsqueda de puntos dentro de la gráfica.
- Filtro en la gráfica.

Trabajos relacionados

En este apartado hablaremos de proyectos similares al realizado, que existen actualmente en el mercado.

6.1. Grafana

Es el trabajo relacionado más famoso de todos, y es empleado para monitorizar todo tipo de datos.

A diferencia de mi proyecto, grafana permite cargar directamente los datos desde una base de datos, lo que es muy útil en empresas que quieran mostrar datos a sus clientes.

Grafana es gratuita (open source)[1], y está basada en cuadros de mandos fácilmente editables , que hacen de ésta una herramienta muy potente, donde el usuario selecciona que datos ver, y como los quiere ver; dentro de las diversas opciones que te permiten seleccionar.

A pesar de ello, los gráficos que ofrece a veces son bastante confusos, y no esta planteado para trabajar con cursores, a pesar de ser un proyecto que admite muchas funcionalidades adicionales, al ser código open source.

6.2. Graphite

Al igual que grafana, graphite es una herramienta open source que permite monitorizar y graficar datos de sistemas informáticos en tiempo real. Destaca por la exactitud del tiempo real con la que ofrece sus datos.

Graphite funciona de la siguiente forma[21]:

- Un demonio en segundo plano, escucha los datos a tiempo real.
- Estos datos se guardan en una base datos.

- Una webapp llamado Django, renderiza los puntos en tiempo real.

6.3. ChartBlocks

Este es un editor muy fácil de usar, que permite realizar gráficos con relativa facilidad.

A diferencia del mi proyecto, y de los dos anteriores, destaca por su facilidad de uso, aunque en uso, no es tan potente como los anteriores.[2]

Permite leer datos de distintas fuentes, incluso de datos en tiempo real y destaca la facilidad de su interfaz.

Conclusiones y Líneas de trabajo futuras

Para concluir con la memoria de este proyecto, presentaré en primer lugar las posibles líneas de trabajo futuras y finalizaré con una conclusión del mismo.

7.1. Líneas de trabajo futuras

A continuación, propongo un par de lineas futuras para continuar con esta aplicación, que no fueron posible realizar por cuestión de tiempo:

- Que la aplicación permita leer datos de otras fuentes, como por ejemplo de una base de datos, para poder realizar y monitorear gráficas en tiempo real.
- Que las ventanas sean escalables, y personalizables, de tal modo, que el usuario pueda hacer pequeñas modificaciones o eliminar ciertos elementos que se presenten en la gráfica.
- Crear una aplicación universal en la red, que te permita guardar el estado actual de una gráfica y poder compartir el enlace. Así como seleccionar archivos y distintas disposiciones de las tramas.
- Migrar la aplicación web a escritorio.

7.2. Conclusiones

En este proyecto, a nivel personal, me he enfrentado a muchos retos.

- Por un lado, me he dado cuenta, que independiente de las tecnologías que conozcas, tienes que adaptarte en cada proyecto a la situación real.

Por ejemplo, no puede implantar un sistema de base de datos, porque el dispositivo no tenía la capacidad de procesamiento necesaria para ello. Y tuve que realizar todo el proyecto en estructuras de datos.

- También me he dado cuenta, de la importancia que han tenido para mi ciertas asignaturas de la carrera, que en principio pensaba que no tenían mucha utilidad práctica. Por ejemplo, he utilizado en un grado elevado los conocimientos adquiridos en la asignatura de procesadores del lenguaje, a la hora de analizar (parsear) el fichero XML o en crear una sintaxis para la búsqueda y los filtros.

  En general pienso que este proyecto tenía un trabajo y dificultad elevado, pues ciertos requisitos eran técnicamente complicados y puntillosos, como pudo ser el filtrado de datos, o la reproducción de la gráfica, por lo que tuve que documentarme e investigar en muchos de ellos varios días sin poder obtener ningún resultado.

  Personalmente pienso que he realizado un buen trabajo con la herramienta de graficado, y que he cumplido satisfactoriamente en un tiempo razonable todos los requisitos propuestos por mi tutor, que en muchos casos se fueron ampliando en el tiempo.

  También agradezco poder ver de primera mano cómo confluyen el mundo de la informática con el mundo industrial, y qué he sido capaz de realizar con mis conocimientos una vez visto el resultado final.

Por último, valoro positivamente la experiencia en la realización de este proyecto; el conocimiento plasmado en el proyecto y espero poder seguir desarrollando proyectos de diversa índole para poder progresar como desarrollador de software.

Bibliografía

[1] ArchWiki. Grafana, 2017. [Internet; consultado 24-junio-2017] https://wiki.archlinux.org/index.php/Grafana.

[2] ChartBlocks. Chartblocks, 2017. [Internet; consultado 24-junio-2017] http://www.chartblocks.com/es/features.

[3] JetBrains. Phpstorm, 2017. [Internet; consultado 13-mayo-2017] https://www.jetbrains.com/phpstorm/.

[4] JetBrains. Webstorm, 2017. [Internet; consultado 22-febrero-2017] https://www.jetbrains.com/webstorm/.

[5] Matrix. Datasheet gtw-mtx, 2017. [Internet; consultado 15-abril-2017] ftp://ftp.matrix.es/mtxm2m/Gateways%20M2M/MTX-GTW/MTX-GTW_Datasheet.pdf.

[6] PHP. Php 5.5, 2017. [Internet; consultado 3-mayo-2017] http://php.net/manual/es/migration55.changes.php.

[7] Sprintometer. Sprintometer, 2017. [Internet; consultado 3-marzo-2017] http://sprintometer.com/.

[8] StackOverFlow. ¿es js un idioma orientado a objetos?, 2017. [Internet; consultado 13-abril-2017] https://stackoverflow.com/questions/107464/is-javascript-object-oriented.

[9] TexMaker. Texmaker, 2017. [Internet; consultado 15-junio-2017] http://www.xm1math.net/texmaker/.

[10] W3Schools. Html5, 2017. [Internet; consultado 15-mayo-2017] https://www.w3schools.com/html/html5_intro.asp.

[11] W3Schools. Xml parser, 2017. [Internet; consultado 3-marzo-2017] https://www.w3schools.com/xml/xml_parser.asp.

[12] W3Schools. Xsd, 2017. [Internet; consultado 22-junio-2017] https://www.w3schools.com/xml/schema_intro.asp.

[13] Wikipedia. Bootstrap, 2017. [Internet; consultado 22-marzo-2017] https://es.wikipedia.org/wiki/Bootstrap_(framework).

[14] Wikipedia. Bus can, 2017. [Internet; consultado 12-marzo-2017] https://es.wikipedia.org/wiki/Bus_CAN.

[15] Wikipedia. Chartjs, 2017. [Internet; consultado 15-marzo-2017] www.chartjs.org/.

[16] Wikipedia. Desarrollo ágil de software, 2017. [Internet; consultado 4-junio-2017] https://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software.

[17] Wikipedia. Ecmascript, 2017. [Internet; consultado 13-abril-2017] https://es.wikipedia.org/wiki/ECMAScript.

[18] Wikipedia. Endianness, 2017. [Internet; consultado 22-abril-2017] https://es.wikipedia.org/wiki/Endianness.

[19] Wikipedia. Git, 2017. [Internet; consultado 3-marzo-2017] https://es.wikipedia.org/wiki/Git.

[20] Wikipedia. Github, 2017. [Internet; consultado 14-abril-2017] https://es.wikipedia.org/wiki/GitHub.

[21] Wikipedia. Graphite (software), 2017. [Internet; consultado 24-junio-2017] https://en.wikipedia.org/wiki/Graphite_(software).

[22] Wikipedia. Hojas de estilo en cascada, 2017. [Internet; consultado 13-abril-2017] https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada.

[23] Wikipedia. Html, 2017. [Internet; consultado 15-mayo-2017] https://es.wikipedia.org/wiki/HTML.

[24] Wikipedia. Javascript, 2017. [Internet; consultado 13-abril-2017] https://es.wikipedia.org/wiki/JavaScript.

[25] Wikipedia. Jquery, 2017. [Internet; consultado 5-marzo-2017] https://jquery.com/.

[26] Wikipedia. Json, 2017. [Internet; consultado 15-abril-2017] https://es.wikipedia.org/wiki/JSON.

[27] Wikipedia. Latex, 2017. [Internet; consultado 15-junio-2017] https://es.wikipedia.org/wiki/LaTeX.

[28] Wikipedia. Metodología scrum, 2017. [Internet; consultado 4-junio-2017] https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software).

[29] Wikipedia. Patrón decorador, 2017. [Internet; consultado 16-mayo-2017] https://es.wikipedia.org/wiki/Decorator_(patr%C3%B3n_de_dise%C3%B1o).

[30] Wikipedia. Single page application, 2017. [Internet; consultado 1-mayo-2017] https://es.wikipedia.org/wiki/Single-page_application.

[31] Wikipedia. Vehículo de guiado autónomo, 2017. [Internet; consultado 2-mayo-2017] https://es.wikipedia.org/wiki/Veh%C3%ADculo_de_guiado_autom%C3%A1tico.

[32] Wikipedia. World wide web consortium, 2017. [Internet; consultado 2-mayo-2017] http://www.w3c.es/.

[33] Wikipedia. Xml, 2017. [Internet; consultado 3-marzo-2017] https://es.wikipedia.org/wiki/Extensible_Markup_Language.