

---

---

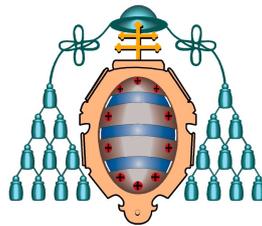
# Técnicas de analítica visual para análisis de la demanda energética en edificios

---

---

Por

PABLO FERNÁNDEZ GONZÁLEZ



Departamento de Ingeniería Eléctrica, Electrónica, de Computadores y Sistemas  
UNIVERSIDAD DE OVIEDO

Un TFM presentado a la Universidad de Oviedo en conformidad con los requisitos del MÁSTER DE INGENIERÍA DE LA AUTOMATIZACIÓN E INFORMÁTICA INDUSTRIAL de la Facultad de Ingeniería.

ENERO 2015



## RESUMEN

**E**l camino para conseguir una mejora de la eficiencia, energética en cualquier ámbito, siempre pasa por medir los datos de consumo e interpretarlos. Es en esta fase de interpretación donde ahonda este trabajo haciendo posible, de una forma muy simple e intuitiva, que el usuario pueda interactuar con toda la información que está visualizando. Gracias a esta interacción, las posibilidades de visualizar diferentes tipos de patrones son casi infinitas.



## DEDICACIÓN Y AGRADECIMIENTOS

**M**e gustaría dedicar este trabajo a los profesores del máster en *Ingeniería de la Automatización e Informática Industrial*, los cuales posibilitaron mi vuelta a los estudios tras ocho años alejado de las instituciones académicas y a Daniel Pérez López, compañero de Laboratorio. Quiero hacer especial hincapié en la persona de Ignacio Díaz Blanco, al ser él quien logró transmitirme la pasión por este campo de la ciencia, haciéndome ver sus grandes posibilidades, no ya sólo a través de su asignatura *Técnicas inteligentes de inspección industrial*, sino también al mostrarme su propuesta de trabajo para el *TFM*. En ella me proponía trabajar con *Pandas de Python*, *HTML*, *JavaScript*, *D3.js* y *Crossfilter.js*, para poder llevar a cabo un proyecto de aplicación Web para visualización y análisis de datos. La idea de aprender todos estos lenguajes, hasta ese momento , completamente nuevos para mí, me resultó irrechazable.



## DECLARACIÓN DEL AUTOR

**D**eclaro que este trabajo ha sido realizado en acuerdo a los requerimientos del Máster de Ingeniería de la Automatización e Informática Industrial de la Escuela Politécnica de Ingeniería de Gijón. Excepto donde sea indicado por referencias específicas en el texto, el trabajo aquí presentado es el propio del candidato. Todos los puntos de vista aquí expresados son los del autor.

FIRMADO: ..... FECHA: .....



## TABLA DE CONTENIDOS

	<b>Pag.</b>
<b>Lista de tablas</b>	<b>ix</b>
<b>Lista de figuras</b>	<b>xi</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Contexto del proyecto . . . . .	1
1.2 Planteamiento del problema . . . . .	6
1.3 Objetivos . . . . .	6
<b>2 Estado del arte</b>	<b>11</b>
2.1 Visualización en Eficiencia Energética . . . . .	11
2.2 Modos y técnicas de visualización . . . . .	15
2.2.1 Visualizaciones 2D . . . . .	16
2.2.2 Visualizaciones 3D . . . . .	18
2.2.3 Múltiples Vistas Coordinadas . . . . .	19
2.3 Tecnologías de visualización . . . . .	19
2.4 Encuadre del proyecto . . . . .	21
<b>3 Métodos y Técnicas</b>	<b>23</b>
3.1 Técnicas . . . . .	23
3.1.1 Tratamiento previo de datos: Python y Pandas . . . . .	24
3.1.2 El estándar Web (Html, CSS y JavaScript) . . . . .	25
3.1.3 D3.js y CrossFilter.js . . . . .	27
3.2 Criterios de diseño . . . . .	29
3.2.1 Criterios funcionales . . . . .	30
3.2.2 Criterios no funcionales . . . . .	32
3.3 La aplicación: plantilla html, archivos y funciones JavaScript . . . . .	33
3.3.1 Resumen del proceso previo realizado Python . . . . .	33
3.3.2 Plantilla de la Aplicación: HTML y CSS . . . . .	35
3.3.3 Archivos utilizados . . . . .	36

## TABLA DE CONTENIDOS

---

3.3.4	Funciones JavaScript y nociones básicas de D3.js . . . . .	37
<b>4</b>	<b>Resultados</b>	<b>49</b>
4.1	Casos de estudio . . . . .	49
4.1.1	Identificación de los consumos más frecuentes . . . . .	50
4.1.2	Identificación de cuando se producen los mayores consumos . . . . .	50
4.1.3	Identificación de cuándo se producen los menores consumos . . . . .	51
4.1.4	Identificación de cuándo se producen los peores factores de potencia y consecuencias . . . . .	52
4.1.5	Identificación de consumos residuales . . . . .	53
4.1.6	Identificación del cambio en la factura al hacer modificaciones en la tarifa horaria . . . . .	54
4.2	Estudio de usuario . . . . .	54
4.2.1	Conclusiones extraídas y sugerencias aportadas por los usuarios . . . . .	56
<b>5</b>	<b>Discusión General</b>	<b>65</b>
5.1	Aportaciones . . . . .	65
5.2	Líneas futuras de trabajo . . . . .	67
<b>A</b>	<b>Cronograma del trabajo</b>	<b>69</b>
<b>B</b>	<b>Presupuesto del trabajo</b>	<b>71</b>
	<b>Bibliografía</b>	<b>73</b>

## LISTA DE TABLAS

<b>TABLA</b>	<b>Pag.</b>
3.1 Estructura de los datos del ejemplo en D3 . . . . .	44
3.2 Tabla de funciones del proyecto. . . . .	48
4.1 Resultados obtenidos del UserStudy . . . . .	55



## LISTA DE FIGURAS

<b>FIGURA</b>	<b>Pag.</b>
1.1 Consumo energético de Europa por sectores . . . . .	2
1.2 Factores que provocan el cambio climático . . . . .	3
1.3 Evolución de las emisiones globales . . . . .	3
1.4 Evolución de los niveles de CO2 en los últimos 400.000 años . . . . .	4
1.5 Tarifas eléctricas EDP del 2016 . . . . .	5
1.6 Captura de la aplicación inspiradora de este trabajo (1) . . . . .	7
1.7 Captura de la aplicación inspiradora de este trabajo (2) . . . . .	7
2.1 Primera visualización de consumos energéticos realizada por ordenador . . . . .	12
2.2 Diferentes representaciones de conjuntos de datos con idénticos parámetros estadísticos	14
2.3 Tipos de visualizaciones 2D tradicionales . . . . .	16
2.4 Tipos de visualizaciones 2D modernas . . . . .	18
2.5 Tipos de visualizaciones 3D . . . . .	19
2.6 Múltiples vistas coordinadas (MVC) . . . . .	20
2.7 Visualización de consumos eléctricos desarrollada con Processing . . . . .	21
3.1 Porcentajes de uso de los lenguajes más usados en el 2015 . . . . .	24
3.2 DataFrame de Pandas . . . . .	25
3.3 Jerarquía de DOM . . . . .	26
3.4 Benchmarks de Rendimiento D3.js . . . . .	28
3.5 Sumario de los criterios de diseño . . . . .	30
3.6 Consideraciones software no funcionales . . . . .	33
3.7 Captura sintetizada de la plantilla HTML del proyecto . . . . .	35
3.8 Capturas de las diferentes partes de la aplicación. . . . .	37
3.9 Captura del <i>Explorador de Soluciones</i> del Visual Studio 2013 . . . . .	38
3.10 Aspecto de la gráfica creada con la función Crear_Histo2D . . . . .	39
3.11 Aspecto de la gráfica creada con la función Crear_Histo . . . . .	40
3.12 Aspecto de la tabla creada con la función Crear_Spreadsheet . . . . .	40
3.13 Aspecto de los sliders creados con la función crear_slider . . . . .	41
3.14 Ejemplo de representación sencilla en D3 . . . . .	47

4.1	Captura que muestra los consumos más frecuentes en ambas facultades. . . . .	50
4.2	Capturas que muestran los consumos más frecuentes de cada facultad . . . . .	57
4.3	Capturas que muestran los mayores consumos de cada facultad . . . . .	58
4.4	Capturas que muestran los menores consumos de cada facultad . . . . .	59
4.5	Capturas de las diferentes partes de la aplicación. . . . .	60
4.6	Capturas del precio kWh de ambas facultades . . . . .	61
4.7	Captura que muestra el consumo total de la facultad de económicas. . . . .	62
4.8	Capturas de consumos residuales . . . . .	63
4.9	Capturas de los costes, en euros, del total de ambas facultades . . . . .	64
5.1	Captura actual de la estructura de gráficos de la aplicación . . . . .	66
A.1	Cronograma del trabajo . . . . .	69
B.1	Presupuesto del trabajo . . . . .	71

## INTRODUCCIÓN

**H**ay pocos temas hoy en día que se puedan considerar de mayor actualidad que el de la *eficiencia energética*, ya sea desde su perspectiva puramente económica, tratar de reducir la factura, como desde la vertiente medioambiental, la reducción de uno de los gases causantes del *efecto invernadero*, el  $CO_2$ . Estas dos vertientes están presentes de distintas formas en los hogares, en las industrias, en los asuntos de estado o, incluso, en las cumbres internacionales. Por todo ello, este capítulo trata de explicar el porqué de la decisión de crear este trabajo, partiendo del contexto en el que se encuentra y acabando en los objetivos marcados para su creación y desarrollo.

## 1.1 Contexto del proyecto

En años recientes, la conciencia social sobre los problemas medioambientales, tales como el incremento de la temperatura o la aceleración en el deshielo de los casquetes polares, ha crecido a escala global. En el futuro, el incremento de la *eficiencia energética* de los sistemas y la reducción de la demanda de energía por parte de los usuarios y empresas, jugará un papel fundamental si queremos poner freno, antes de que sea demasiado tarde, al *Cambio Climático*. Como se puede apreciar en la figura 1.7, el consumo energético europeo, en hogares e industria, suponen más de la mitad (58%) de los consumos totales.

Es conveniente aquí hacer un inciso para explicar brevemente lo que se entiende por *cambio climático* y *eficiencia energética*. "Se llama cambio climático a la modificación del clima con respecto al historial climático a una escala global o regional. Tales cambios se producen a muy diversas escalas de tiempo y sobre todos los parámetros meteorológicos: temperatura, presión atmosférica, precipitaciones, nubosidad, etc. En teoría, son debidos tanto a causas naturales

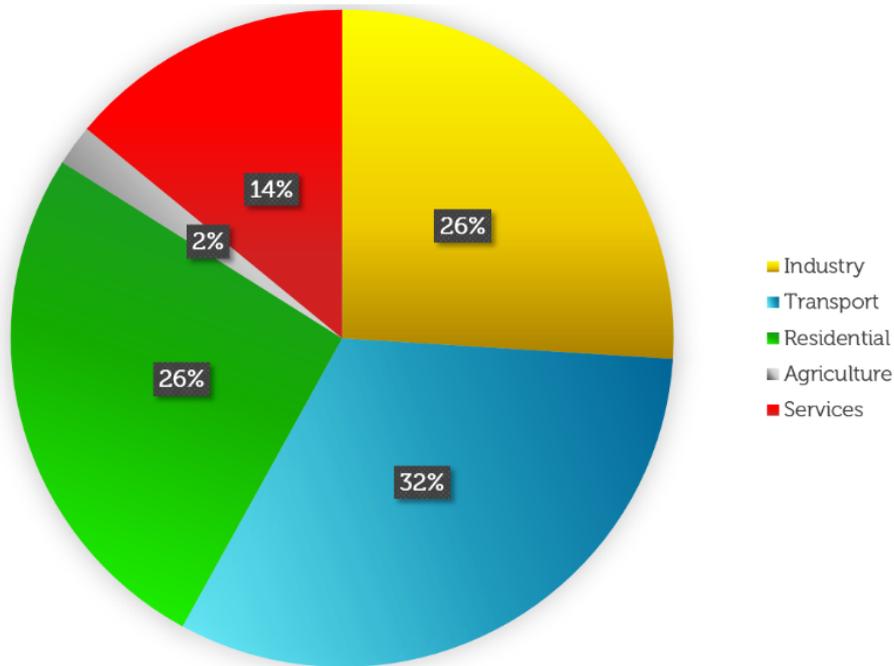


FIGURA 1.1. Consumo energético de Europa por sectores (datos del año 2012. Fuente: Comisión europea - Eurostat)

(Crowley y North, 1988) como antropogénicas (Oreskes, 2004)"[30]. Este término suele usarse de manera poco apropiada utilizándose como sinónimo de *calentamiento global*. La Convención Marco de las Naciones Unidas sobre el Cambio Climático usa este término solo para referirse al cambio por causas humanas:

*"Por cambio climático se entiende un cambio de clima atribuido directa o indirectamente a la actividad humana que altera la composición de la atmósfera mundial y que se suma a la variabilidad natural del clima observada durante períodos comparables". (Artículo 1, párrafo 2)*

En cuanto a la *eficiencia energética*, se podría resumir como la relación entre productos y servicios finales obtenidos y la cantidad de energía consumida para lograrlo (ecuación 1.1).

$$(1.1) \quad \text{Eficiencia\_Energética} = \frac{\text{Productos\_y\_Servicios}}{\text{Consumo\_Energía}}$$

Una vez explicado esto, podemos extraer relaciones directas entre ambos términos. El Cambio Climático, tal y como lo describe Naciones Unidas, es el producido por acción del hombre al usar *combustibles fósiles*<sup>1</sup> para extraer de ellos su energía, ya sea para el transporte, generar

<sup>1</sup>Aquellos que proceden de la biomasa producida hace millones de años, que pasó por grandes procesos de transformación hasta la formación de sustancias de gran contenido energético como el carbón, el petróleo o el gas natural. Debido a su procedencia biológica, su componente principal es el *Carbono*.

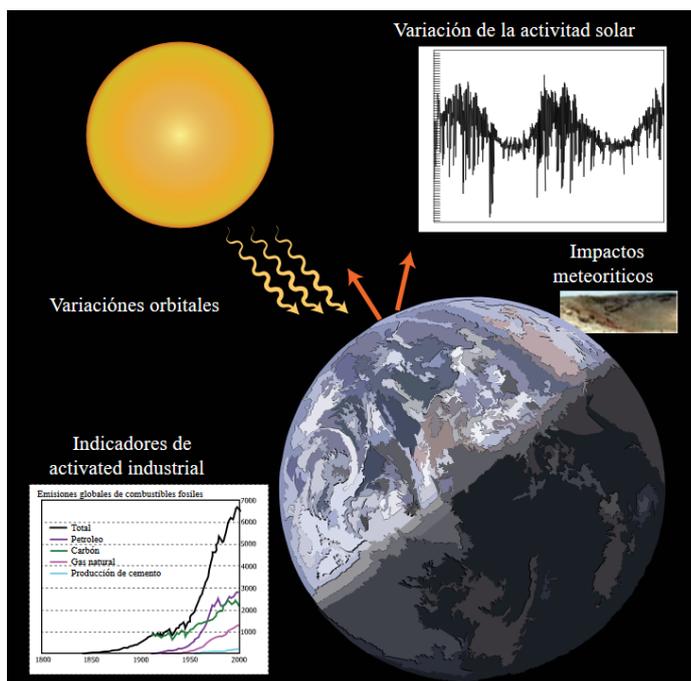


FIGURA 1.2. Esquema ilustrativo de los principales factores que provocan los cambios climáticos actuales de la Tierra. La actividad industrial y las variaciones de la actividad solar se encuentran entre los más importantes. Fuente - Wikipedia

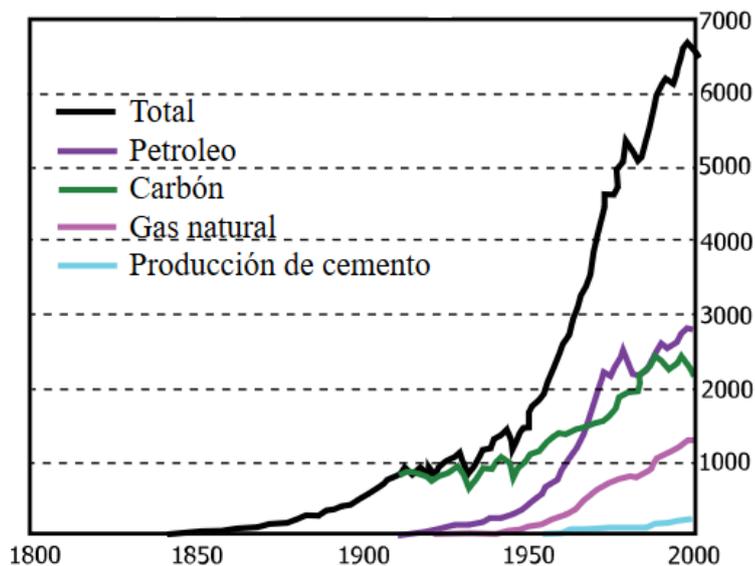


FIGURA 1.3. Detalle de la figura anterior en la que se aprecia mejor la evolución de las emisiones, por combustibles fósiles, durante los últimos 200 años. Fuente - Wikipedia

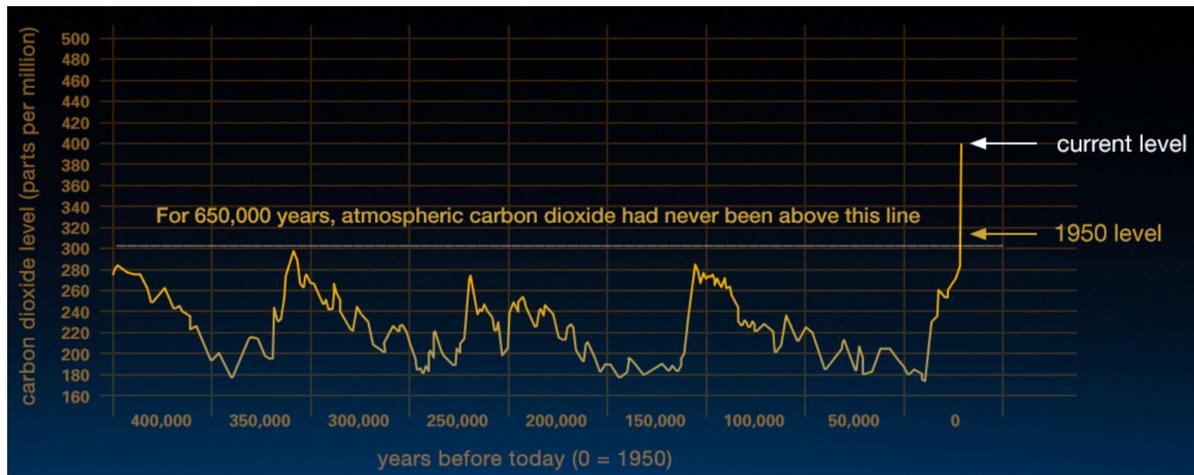


FIGURA 1.4. Evolución de los niveles de CO<sub>2</sub> en partes por millón desde hace 400.000 años. Está basado en la comparación de las muestras atmosféricas contenidas en los núcleos de hielo y las medidas directas realizadas recientemente (Crédito - Vostok ice core data/J.R. Petit et al.; NOAA Mauna Loa CO<sub>2</sub> record.)

electricidad o calor. Estos combustibles liberan su energía al ser quemados a la vez que emiten gases como el conocido *Dióxido de Carbono* (CO<sub>2</sub>), uno de los principales causantes del *efecto invernadero*<sup>2</sup> y, por consiguiente, del *calentamiento global* y *Cambio Climático*. La figura 1.4, extraída de la página web de la NASA [16], muestra un incremento del 30% en los niveles de CO<sub>2</sub> a partir del año 1950.

Es aquí donde entra en juego la relación con la *eficiencia energética* al ser su mejora fundamental para reducir los niveles de CO<sub>2</sub>, cuya relación con la energía eléctrica producida en las centrales dedicadas a este fin es *lineal*; aproximadamente, esta relación es de  $1 \text{ kWh} - 0,649 \text{ Kg de CO}_2$  [11]. Se puede decir por ello, que el desafío de frenar el Cambio Climático al que se enfrenta la Humanidad, pasa por ser conscientes de la energía que se gasta. Sólo sabiendo cuando, donde y cuanto se consume, se podrá llegar a tener la *conciencia energética* tan necesaria para frenarlo. Esto no sólo tiene los beneficios medioambientales derivados de reducir los niveles de CO<sub>2</sub><sup>3</sup>, sino también económicos.

El coste de la factura eléctrica, ya sea en el hogar, en la industria o un edificio público, viene dado, principalmente, por *dos* factores, la *potencia máxima contratada* y la *energía consumida*. El primero de ellos hace mención a la máxima potencia que el contratante puede disponer sin que se

<sup>2</sup>Es un "proceso en el que la radiación térmica emitida por la superficie planetaria es absorbida por los gases de efecto invernadero (GEI) atmosféricos y es reirradiada en todas las direcciones. Ya que parte de esta reirradiación es devuelta hacia la superficie y la atmósfera inferior, resulta en un incremento de la temperatura superficial media respecto a lo que habría en ausencia de ellos"[31]

<sup>3</sup>A pesar de ser el dióxido de carbono el más conocido del grupo de los gases causantes del efecto invernadero, hay otros que se encuentran en menor medida como son el metano, ozono, óxido de nitrógeno y los CFC.

## Precios Electricidad

Tipo Suministro	Nombre	Precio Potencia (€/kW mes)			Precio Consumo (€/kWh)		
		P1	P2	P3	P1	P2	P3
Potencia <= 10 kW	Precio Luz EDP 1	3,503627	N/A	N/A	0,124617	N/A	N/A
	Precio Luz EDP 1 DH	3,503627	N/A	N/A	0,146416	0,064569	N/A
Potencia > 10 kW y <= 15 kW	Precio EDP 2	3,755160	N/A	N/A	0,149532	N/A	N/A
	Precio Luz EDP 2 DH	3,703737	N/A	N/A	0,177331	0,078088	N/A
Potencia > 15 kW	Precio Luz EDP 3 Periodos	3,394071	2,036455	1,357616	0,130152	0,111372	0,085630

FIGURA 1.5. Tarifas eléctricas de EDP del 2016 para potencias contratadas de hasta 15kW (precios del mercado libre). Fuente - EDP Energía

corte<sup>4</sup> o penalice<sup>5</sup> su servicio, mientras que la segunda habla por sí sola; es la totalidad de energía consumida, normalmente en kWh, en un periodo determinado de tiempo, generalmente mensual. En la figura 1.5, que muestra las tarifas de EDP<sup>6</sup> del año 2016 cuya potencia sea inferior a 15 kW, se pueden apreciar estos dos aspectos como *Precios Potencia (kW/mes)* y *Precios Consumo (€/kWh)*. Este último ofrece además, según tarifa, diferentes precios del kWh en función del tramo horario (representado como P1, P2 y P3).

Es obvio por tanto, que conocer el *pico máximo de consumo* que se puede dar en un momento dado, es un factor determinante a la hora de reducir la factura; por un lado porque permite ajustar la tarifa contratando la potencia justa, sin pagar de más por algo que no se llegaría a usar y, por otro lado, si se conoce *cuándo* se da, a que horas determinadas, podría ser posible llegar a ajustarlas a las de *precio/kWh* reducido, dando lugar a un ahorro sustancial.

Ser capaces de conocer nuestros consumos y tendencias, ya sea en el hogar, la industria o edificios públicos, requiere el uso de una serie de técnicas de visualización que faciliten la toma de decisiones de una forma informada. Visualizar esta información de manera que sea entendible y contextualmente apropiada puede ayudar a cerrar la brecha entre meros datos y esa toma de decisiones, con conocimiento, que permita un óptimo uso de la energía que nos brinde beneficios, tanto a nivel económico como medioambiental.

<sup>4</sup>El servicio se puede interrumpir o cortar por dos razones; que la protección instalada en el lugar del contratante haya llegado a su límite y se desconecte por protección o que el propio suministrador la bloquee al alcanzarse el tope máximo contratado.

<sup>5</sup>En las grandes industrias, debido a que no es viable un corte del suministro bajo ningún concepto

<sup>6</sup>Siglas de *Energía de Portugal*, uno de los principales grupos eléctricos de Europa, y el mayor de Portugal

## 1.2 Planteamiento del problema

Visualizar los datos de consumo de forma que se cumpla con los principios básicos de visualización expuestos anteriormente (que sean *comprensibles y contextualmente adecuados*) es de vital importancia en la búsqueda de la eficiencia energética. El problema, actualmente, es que las herramientas disponibles para el *Análisis del Consumo Energético* rara vez van más allá de una mera representación de los datos de consumo y, más raro aún, que ofrezcan interacción con ellos.

El objetivo de este trabajo es demostrar la utilidad de ña analítica visual en la mejora de la eficiencia energética por lo que, para ello, se creará una herramienta novedosa que sea capaz de generar esa conciencia energética en los usuarios, ya sean familias o empresas, que permita obtener los beneficios económicos tan ansiados y que, al mismo tiempo, ayude a mejorar el dañado ecosistema. Esto sólo es posible encendiendo la chispa de la curiosidad o estimulando una reacción emocional que haga que el usuario se identifique con la herramienta, que quiera usarla y sacarle partido.

Para conseguir esto, la herramienta no sólo debe mostrar datos de consumos de forma que puedan ser interpretados incluso por gente neófita en la materia, sino que también es necesario dotarla de interacción de forma que ésta y usuario se retroalimenten. Brindar la posibilidad al usuario de experimentar, de *"jugar con los datos"* de forma fluida, con baja latencia, eleva el nivel de cognición y entendimiento de estos, de forma que pasen de ser simples datos a conocimiento adquirido.

## 1.3 Objetivos

Esta aplicación, a diferencia de las actuales, tendrá que poder ser usada en la mayor cantidad de dispositivos posible, sin importar su sistema operativo o potencia, posibilitando que se tenga acceso a ella *a cualquier hora y en cualquier lugar*. Parece claro por tanto que la mejor solución es crear una *aplicación web*.

Si bien no se pretende crear una aplicación idéntica, sí que se tratará de que sea similar en estética, facilidad de uso y fluidez, a la desarrollada por dos expertos en materia de visualización como son M. Bostock<sup>7</sup> y J. Davies<sup>8</sup>; juntos, y para mostrar las bondades de la librería *CrossFilter*[7] a la hora de trabajar con múltiples vistas coordinadas, crearon una herramienta de análisis del tiempo de retraso, en la llegada, de todos los vuelos realizados en Estados Unidos en el año 2001. Las figuras 1.6 y 1.7 muestran capturas de ésta.

En ella el usuario puede seleccionar *"una porción"* en cualquiera de los histogramas, y el resto,

---

<sup>7</sup>Doctorado en Computer Science por la universidad de Stanford y creador de D3, librería JavaScript para producir visualizaciones de datos dinámicas e interactivas en los navegadores web.

<sup>8</sup>Licenciado en Computer Science por la universidad de Cambridge.

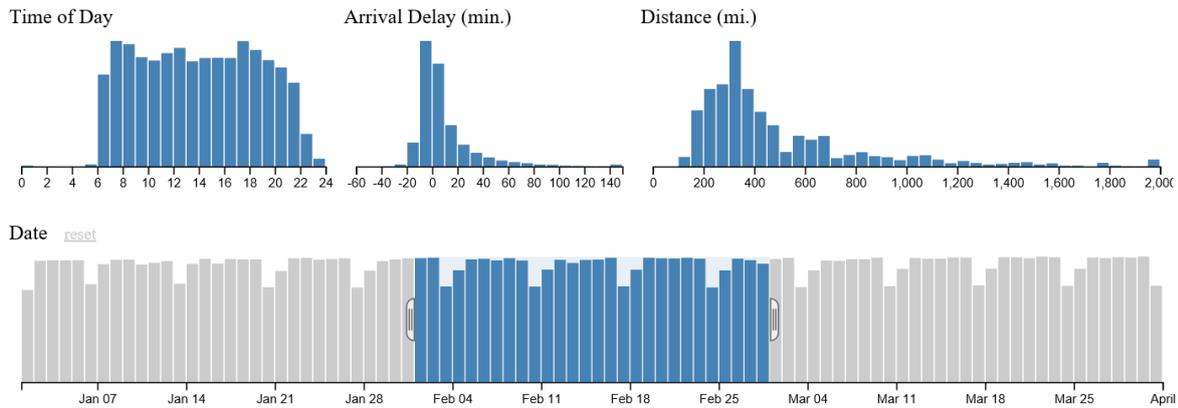
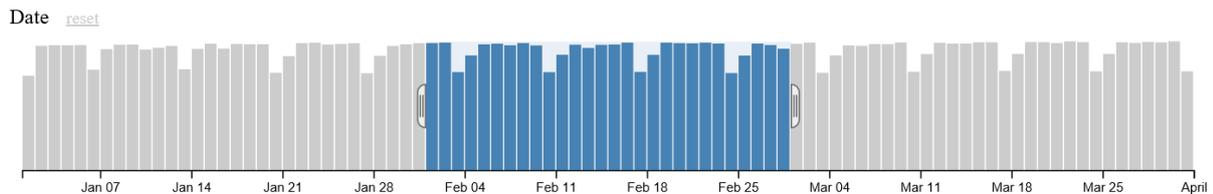


FIGURA 1.6. Captura de la aplicación inspiradora de este trabajo.

al igual que la tabla mostrada en la figura 1.7, se actualizan en tiempo real mostrando sólo los datos que cumplen esos criterios seleccionados. Esto, que no es más que una descripción un tanto coloquial de la forma de proceder de las *MCV* ( Multiple Coordinated Views ) hace perfectamente entendible su gran potencial.

En el caso que nos ocupa, los histogramas a mostrar serán *siete* distribuidos de manera que *dos* de ellos, los que muestran magnitudes eléctricas, *Potencia Activa* y *Factor de Potencia*, se colocarán arriba, ocupando todo el ancho de la página web, y los *cinco* encargados de aportar el



February 28, 2001 71,818 of 231,083 flights selected.

11:59 PM	LAS	LAX	236 mi.	+139 min.
11:58 PM	PHX	SAN	304 mi.	+83 min.
11:49 PM	SJC	PDX	569 mi.	+172 min.
11:42 PM	PHX	OAK	646 mi.	+97 min.
11:41 PM	PHX	LAX	370 mi.	+73 min.
11:40 PM	PHX	ONT	325 mi.	+92 min.
11:35 PM	PHX	ONT	325 mi.	+16 min.
11:25 PM	ONT	OAK	361 mi.	+75 min.
11:24 PM	STL	MDW	251 mi.	+22 min.
11:22 PM	LAX	SLC	590 mi.	+175 min.

FIGURA 1.7. Captura de la aplicación inspiradora de este trabajo.

resto de datos, ya sean temporales o de ubicación, irán debajo. La lista con todos los histogramas es la siguiente:

- Potencia activa en kW
- Factor de potencia
- Hora del día
- Día de la semana
- Mes de año
- Festivo / No festivo
- Edificio (en este caso se referirá al total de los que forman la universidad)

Además de estos, se creará también un histograma en forma de *scatter*<sup>9</sup> que mostrará, mediante círculos de diferentes colores, la cantidad de mediciones que tienen similares valores eléctricos (Potencia y Factor de potencia), de forma que el usuario vea de un simple vistazo cual es la combinación más repetida a lo largo de un año o, en caso de selección previa por parte de este, el intervalo de tiempo seleccionado.

Como mejora de la aplicación de referencia se desea añadir, a cada histograma, el eje vertical, ya que aporta una valiosa información al usuario sobre el número de muestras que contiene cada barra del histograma que está visualizando. Como, a diferencia de los ejes horizontales, estos sí cambiarán sus escalas con cada selección que el usuario realice ( se harán más pequeñas a medida que se vayan seleccionado condiciones temporales más específicas), deberán ser auto escalados. Esto permite poner en contexto al usuario en todo momento sobre la cantidad de datos que esta visualizando.

Para expandir las posibilidades de esta herramienta de visualización y análisis de datos, se desea poder mostrar valores agregados, tanto de costes en euros como de energía total consumida en kWh y kg de CO2 emitidos, redibujando estos nuevos histogramas en el mismo lugar de los anteriores, de forma que el diseño de la página no cambie ni necesite hacer scroll<sup>10</sup> por parte del usuario. Por tanto, lo que se obtendría, pasaría de ser histogramas que muestran el número de valores que hay dentro de cada grupo (barra), a la energía, costes en euros o kilogramos de CO2 generados en cada uno de ellos.

---

<sup>9</sup>Tipo de representación matemática que usa coordenadas cartesianas para mostrar, típicamente, la correlación entre dos variables de un conjunto de datos. Se puede incrementar el número de variables a cuatro si codificamos en tamaño y color cada uno de los puntos.

<sup>10</sup>"Se denomina scroll, desplazamiento, rollo o voluta al movimiento en 2D de los contenidos que conforman el escenario de un videojuego o la ventana que se muestra en una aplicación informática (por ejemplo, una página web visualizada en un navegador web)" - Wikipedia[32].

Como último añadido, se decide incorporar una herramienta que haga posible modificar los precios/hora de cada una de las *veinticuatro* horas del día, partiendo de los valores actuales contratados. Al usarla junto con los datos agregados de coste en euros, se obtiene lo que podría denominarse "*ingeniería tarifaria*", ya que abre un nuevo y amplio abanico de estudios. Se podría analizar el ahorro o gasto en exceso que provoca cualquier variación sobre los precios actuales, incluso acotando ciertos límites temporales o de ubicación, para estudios más detallados.

Para visualizar tanto los datos agregados de consumo de energía o costes como la herramienta de modificación de tarifas, será necesario pulsar el botón correspondiente. También se añadirán botones para visualizar en la tabla (similar en diseño a la de la figura 1.7) los *8 mayores o menores registros*, según la preferencia, en función de la *potencia activa* o el *factor de potencia* del conjunto de datos seleccionado.

La suma de todas estas posibilidades a disposición del usuario lleva la aplicación a un nivel superior de análisis de datos de consumo energético. Esto da pie a un conocimiento total de los patrones de consumo, lo que provoca que se alcancen los más altos niveles cognitivos, al crear conocimiento en el usuario a través de la asimilación de modelos tras un corto periodo de interacción, a su vez, propicia el despertar de esa *conciencia energética* tan necesaria, principal objetivo de este trabajo.



## ESTADO DEL ARTE

La visualización del consumo de energía eléctrica está considerada como una de las más importantes formas de motivar a los usuarios al uso eficiente de esta. Por esta razón, a lo largo de los últimos cuarenta años se ha llevado a cabo una búsqueda incansable de nuevas técnicas que posibiliten un mejor modo de llevarla a cabo. En este capítulo se hará un recorrido a través de estas técnicas que se han utilizado y utilizan a la hora de mostrar datos de consumo de energía eléctrica; desde el envío de postales a familias para informarles de su consumo diario a principios de los 70 hasta las vistosas representaciones 3D que podemos ver hoy en día.

### 2.1 Visualización en Eficiencia Energética

Visualizar del consumo de energía es una área de investigación multidisciplinar que abarca ingeniería de software, eléctrica y mecánica. Por esta razón, el diseño de estas visualizaciones no es en absoluto trivial desde la perspectiva de ingeniería de software; más aun teniendo en cuenta que no hay unos criterios claros que sirvan de guía para esta aplicación de dominio crítico.

A pesar de los factores teóricamente motivacionales como son los económicos y los medioambientales, en la práctica se trata de un reto de motivar a los usuarios a hacer un uso eficiente de la energía. Esto se debe principalmente a la falta de concienciación o conocimiento de sus consumos de energía de una forma precisa, a la incapacidad para discernir medidas completas que deben tomarse para conservarla y también a la renuncia a sacrificar las comodidades de su vida diaria en aras del ahorro energético.

La visualización ha sido largo tiempo considerada como una forma viable de motivar a sus

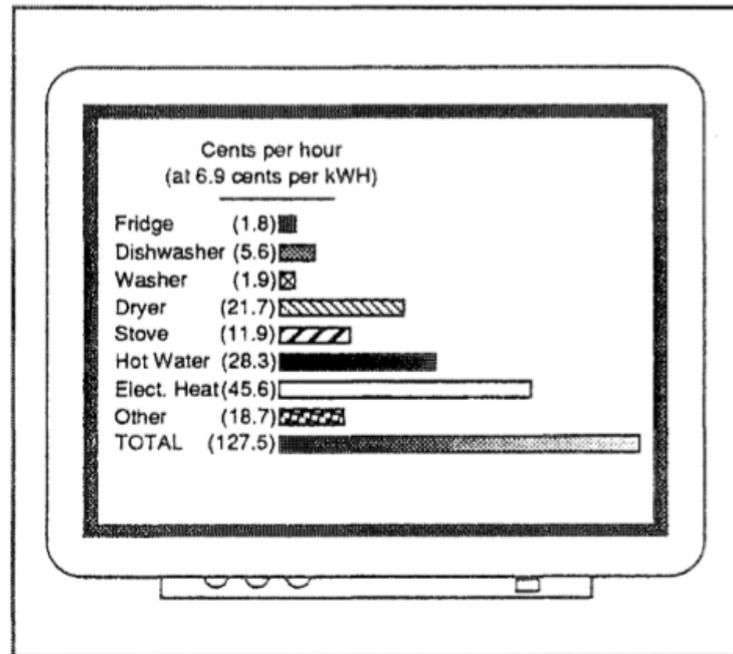


FIGURA 2.1. Primera visualización de consumos energéticos realizada por ordenador en el año 1992, el "Residential Electricity Cost Speedometer (RECS)". Logró reducir el consumo un 12,9%.

usuarios al uso eficiente de la energía al proveerles de la capacidad de monitorizar y, por lo tanto, controlar su consumo de energía. El óptimo diseño de estas visualizaciones es considerado crítico para alcanzar esta meta y generar ahorros en la mayor cantidad de usuarios [1]. A pesar del creciente interés y esfuerzo en esta área, no existen estudios concienzudos que evalúen el estado actual y proporcionen unos criterios claros que guíen el diseño de visualizaciones efectivas.

Los primeros esfuerzos en el ámbito de la visualización datan de principios de la década de los setenta cuando el psicólogo Ronald Bittle<sup>1</sup> y su equipo comenzaron a enviar, diariamente a hogares, tarjetas que informaban de su consumo energético diario. Esta simple medida consiguió una reducción en el consumo que osciló entre el 1% y el 9% [9].

En 1978, una investigación usó grandes posters para mostrar el consumo de energía en una institución médica [3], lo que también provocó un descenso en el consumo de esta. Más tarde, en 1992, las visualizaciones pasaron a un nuevo nivel con el uso de "Residential Electricity Cost Speedometer (RECS)" una herramienta de monitorización computerizada que logró reducir el consumo un 12,9% [21].

Posteriormente, a finales de los años noventa, los datos de consumos energéticos eran visual-

<sup>1</sup>Doctor en Psicología y profesor adjunto en la Universidad Carbondale, Illinois

izados en ordenadores personales [12]. Aunque el *feedback* continuo en el hogar emergió en 1979 [6], la retroalimentación continua, con baja latencia, no fue introducida hasta finales de la década de los noventa, concretamente en el año 1999 [24].

Más recientemente, las visualizaciones en tiempo real se empezaron a representar tanto en 2D como en 3D a la vez que se posibilitó su uso desde cualquier dispositivo, ya fueran teléfonos móviles, tabletas u ordenadores personales al hacerlas *aplicaciones web*. Todo esto contribuyó enormemente a mejorar la accesibilidad.

Otra área de interés ha sido cómo combinar el arte y la tecnología en la visualización para conseguir una mejor comprensión, por parte del usuario, de los datos mostrados. Algunas investigaciones involucran psicólogos para aprovechar aspectos del comportamiento de la psicología humana para incrementar la sostenibilidad energética<sup>2</sup>. En otra investigación, las visualizaciones fueron vinculadas a factores psicológicos humanos a través del impacto medioambiental para conseguir una mejora en la conservación de la energía.

El estadístico estadounidense Edward Tufte<sup>3</sup>, pionero en el campo de la visualización de datos y defensor del minimalismo en la representación gráfica de estos y eliminación de todo tipo de atributos que estorben su comprensión, ha sido el que, sin ningún género de duda, ha revolucionado este campo. "Propugna un estilo sobrio en el que prime la información sobre el adorno. Así, ha acuñado términos como *chartjunk* (basura gráfica) para referirse a aquellos elementos no esenciales de una gráfica que complican su lectura y del *ratio tinta-información*, que mide la cantidad de la primera que es preciso utilizar para representar efectivamente un determinado conjunto de datos"<sup>4</sup>. Propugna que la excelencia a la hora de mostrar gráficos consiste en *comunicar ideas complejas con claridad, precisión y eficiencia*. Por ello, una pantalla gráfica debe [22]:

- mostrar los datos
- inducir al usuario a pensar sobre la "propia sustancia" de ellos más que en el diseño o tecnología empleada
- evitar distorsionar lo que los datos tienen que decir
- presentar muchos números en un pequeño espacio
- hacer coherentes grandes conjuntos de datos

---

<sup>2</sup>También llamada sustentabilidad energética es definida por el World Energy Council (WEC), como el equilibrio entre tres dimensiones principales: la seguridad energética, la equidad social, y la mitigación del impacto ambiental (fuente - Wikipedia).

<sup>3</sup>Profesor emérito de la Universidad de Yale, en la que dictó cursos sobre evidencia estadística y diseño de información y de interfaces. Es autor de varios libros de referencia en este campo, siendo su obra cumbre "*The Visual Display of Quantitative Information*" cuya primera edición data del año 1984 y su segunda y última del 2002.

<sup>4</sup>fuente - Wikipedia

- estimular al ojo a comparar diferentes partes de los datos
- revelar los datos a diferentes niveles de detalle
- servir a un propósito razonablemente claro
- estar estrechamente integrada con la descripción estadística y verbal de un conjunto de datos

(a)

I		II		III		IV	
X	Y	X	Y	X	Y	X	Y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

$N = 11$   
 mean of X's = 9.0  
 mean of Y's = 7.5  
 equation of regression line:  $Y = 3 + 0.5X$   
 standard error of estimate of slope = 0.118  
 $t = 4.24$   
 sum of squares  $X - \bar{X} = 110.0$   
 regression sum of squares = 27.50  
 residual sum of squares of Y = 13.75  
 correlation coefficient = .82  
 $r^2 = .67$

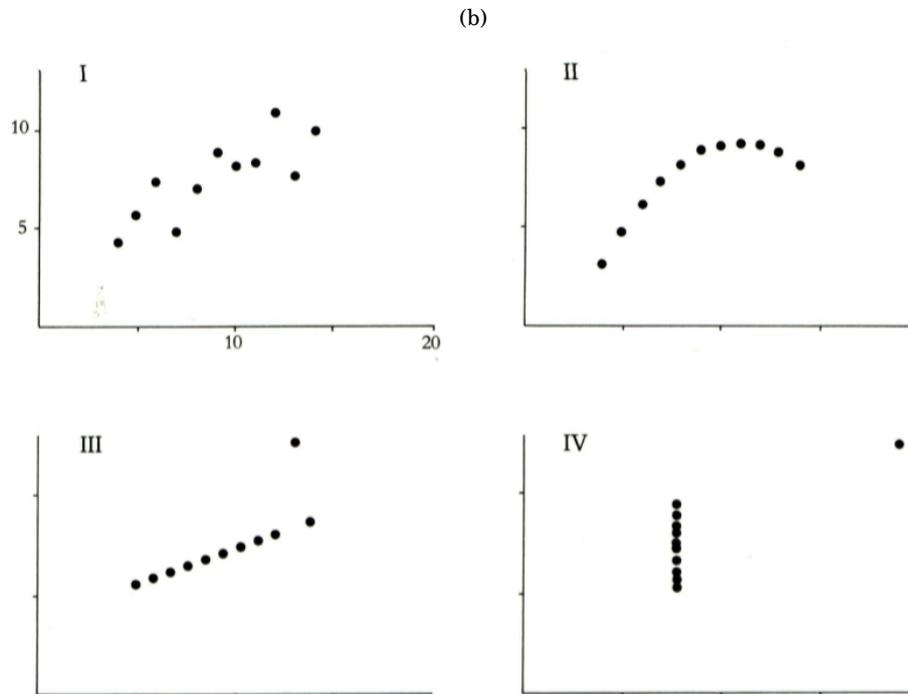


FIGURA 2.2. Imágenes extraídas del libro "The visual display of quantitative information" de Edward Tufte); (a) Cuatro conjuntos de datos diferentes que tienen los mismos parámetros estadísticos; (b) Representación, mediante scatter bivariable, de los 4 conjuntos.

La figura 2.2 muestra el "*cuarteto de Anscombe*" e ilustra de forma elocuente la incapacidad de la estadística descriptiva básica (medias, varianzas y covarianzas) para la descripción de los datos, cuando una simple visualización demuestra que son situaciones completamente distintas (imágenes extraídas del libro de Tufte "The visual display of quantitative information").

La última tendencia en el área de la visualización viene de la mano de las Smart Grids<sup>5</sup>, las cuales se centran en el uso intensivo de la tecnología para hacer frente a los retos de consumo y conservación de la energía. Una de las siete áreas del modelo conceptual de Smart Grid [14] creado por el NIST<sup>6</sup> es la del cliente y una de sus responsabilidades es informarles y hacer que la energía gastada en los hogares se haga de la manera más eficiente posible. Mientras esta área del cliente cubre el ciclo completo de la energía, la investigación de cómo visualizar correctamente estos datos todavía no ha alcanzado su madurez, siendo su búsqueda, por tanto, el camino a seguir en futuros trabajos.

## 2.2 Modos y técnicas de visualización

El papel fundamental de la visualización es proporcionar "*feedback*" sobre el consumo de energía a los usuarios. Debido a que no todos los casos son iguales, existen varios modos que se clasifican, en base a su granularidad<sup>7</sup> con respecto a:

- tiempo
- escala
- información
- técnicas

Los modos basados en la información de la energía consumida podrían ser de la forma de kilovatios-hora (kWh), coste en euros (€), kilogramos de CO<sub>2</sub> emitidos a la atmósfera, etc. Estos modos podrían, a su vez, ser conmutados a otros basados en periodos de tiempo tales como horas, días, meses o años, o basados en escala, como por habitación, planta, edificio, etc. Todos

---

<sup>5</sup>También conocidas como redes inteligentes, se asocia a menudo con el concepto de medidores inteligentes capaces de ofrecer una facturación detallada por franjas horarias, lo que permitiría a los consumidores no solo el elegir las mejores tarifas de entre las diferentes empresas eléctricas, sino también discernir entre las horas de consumo, lo que a su vez permitiría un mejor uso de la red. Este sistema también permitiría mapear con más precisión el consumo y anticipar mejor las necesidades futuras a nivel más local. Fuente - Wikipedia

<sup>6</sup>Siglas de National Institute of Standards and Technology, es el Instituto Nacional de Normas y Tecnología. Entre 1901 y 1988 llamada Oficina Nacional de Normas (NBS por sus siglas del inglés National Bureau of Standards), es una agencia de la Administración de Tecnología del Departamento de Comercio de los Estados Unidos. La misión de este instituto es promover la innovación y la competencia industrial en Estados Unidos mediante avances en metrología, normas y tecnología de forma que mejoren la estabilidad económica y la calidad de vida.

<sup>7</sup>La granularidad de los datos se refiere al tamaño en el cual los campos de los datos son subdivididos. Fuente - Wikipedia.

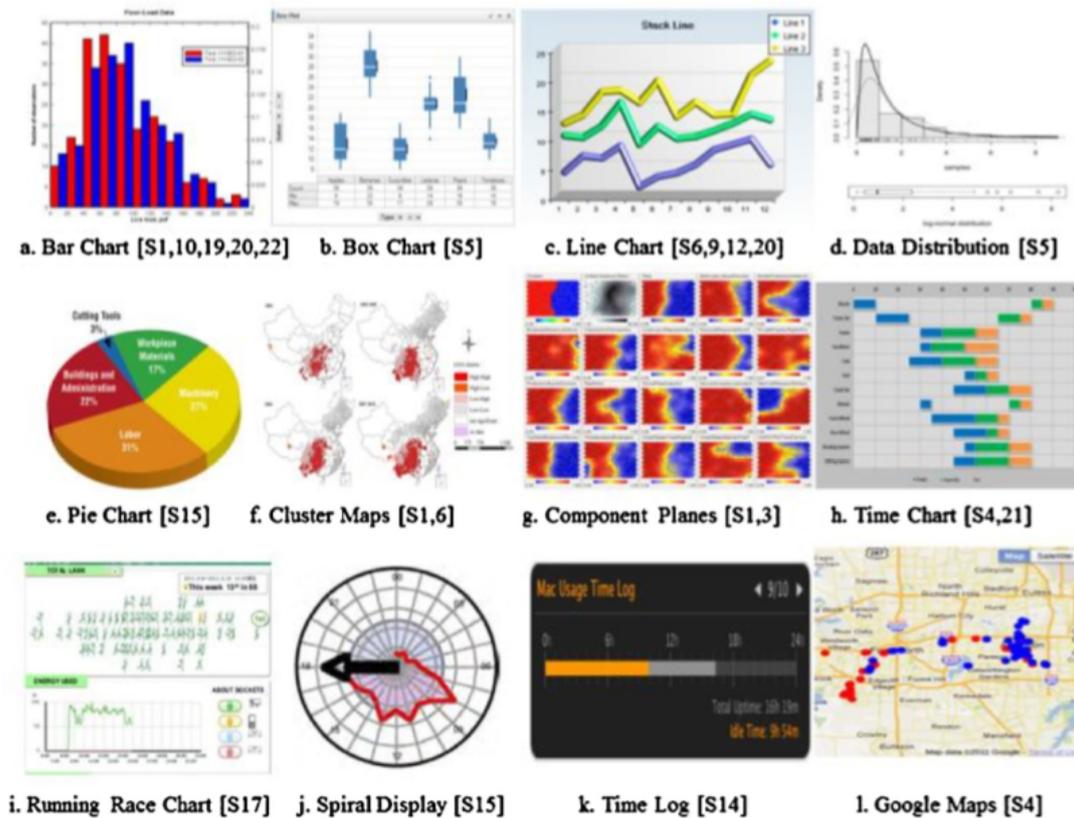


FIGURA 2.3. Tipos de visualizaciones 2D tradicionales. Fuente - Design criteria for visualization of energy consumption: A systematic literature review [15].

estos modos suelen representarse en forma de gráficos de barras debido a su fácil y directa interpretación.

Los modos de visualización basados en técnicas pueden llegar a ser más flexibles a la hora de cambiar entre diferentes representaciones gráficas buscando, con ello, atraer al usuario y sorprenderle. Estas técnicas de visualización se pueden dividir en dos grandes grupos, *visualizaciones 2D* y *visualizaciones 3D*, las cuales, a su vez, pueden crearse desde un enfoque moderno o tradicional (cerca de un 95% de los trabajos en este campo han usado varias de estas técnicas) [15].

### 2.2.1 Visualizaciones 2D

La visualización en dos dimensiones es la forma más común de representar información cuantitativa y esto es, en parte, gracias a su mejor comprensión (el 90% de las visualizaciones de estudios y trabajos sobre consumo de energía han sido desarrollados en 2D [15]). Dentro de este grupo están las visualizaciones tradicionales, las cuales comprenden gráficos básicos y avanzados. Un gráfico básico puede ser realizado desde una hoja de cálculo tipo Excel y ejemplos de este son los

diagramas de:

- barras (bar charts)
- caja (box charts)
- líneas (line charts)
- tarta (pie charts)

Por otro lado, la visualización de gráficos avanzados requiere del uso de herramientas especiales para su creación. Ejemplos de ellos son:

- time charts
- geo charts (el usado por Google Maps)
- self organizing maps (SOM)
- cluster maps
- spiral display

En la figura 2.3 se pueden ver ejemplos de representaciones 2D tradicionales, las cuales son las representaciones 2D más utilizadas cuando se trata de visualizar consumos de energía (se usan un 86% de las veces [15]).

En cuanto a las visualizaciones modernas, se quiere ir un paso más allá añadiendo interacción, más información y mejorando el diseño artístico y la estética. Surgen de los requisitos de diseño actuales para una buena y efectiva visualización; estos requisitos son:

- comprensibilidad
- atracción
- situación periférica (que proporcione varios gráficos o puntos de vista diferentes para ayudar a que el usuario se ubique)
- diseño y estética que fomenten el respeto por el medio ambiente (eco-friendly)

La figura 2.4 muestra ejemplos de este tipo de representaciones 2D modernas, las cuales son usadas en un 45% de los estudios y trabajos de este campo de investigación [15]. En ellas, se pueden ver gráficos basados en la dispersión y contracción de objetos como son el *Hive Design*, *Phyllotaxis Design* y *Pinwheel Design* (*diseño en nido*, *filotaxis*<sup>8</sup> o *molinillo* respectivamente),

---

<sup>8</sup>Disposición que presentan las hojas en el tallo de una planta. Fuente - Wikipedia.

basados en el plano de una casa como el *Red Spots*, en el numero de árboles necesarios para compensar las emisiones de CO<sub>2</sub>, etc.

### 2.2.2 Visualizaciones 3D

Las visualizaciones 3D, las cuales empezaron a usarse en la década de los ochenta, son más realistas y atractivas para el cerebro humano. Por contra, muestran problemas de percepción de las distancias como la profundidad, y pueden resultar confusas para determinados usuarios si no han sido previamente enseñados. Se clasifican según su enfoque, al igual que las visualizaciones 2D, en tradicionales y modernas.

Se consideran visualizaciones tradicionales a las típicas que incluyen gráficos de las hojas de cálculo como los *Chloropeth* y *Hit Maps* o las variaciones de los gráficos 2D como los de barras, tarta, línea, etc..., mientras que las visualizaciones 3D modernas pueden incluir *mapeo de objetos 3D*<sup>9</sup> e interfaces de usuario.

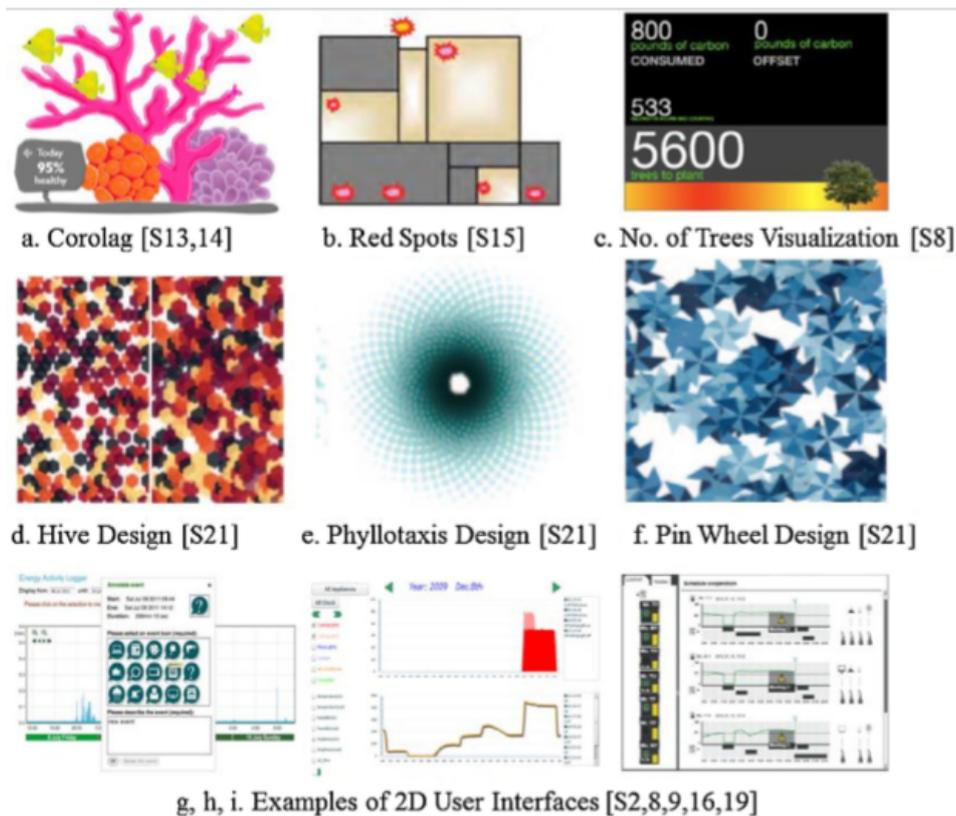


FIGURA 2.4. Tipos de visualizaciones 2D modernas. Fuente - Design criteria for visualization of energy consumption: A systematic literature review [15].

<sup>9</sup>Consiste en aplicar imágenes o fotos en la superficie de objetos 3D de forma que, por ejemplo, un hexaedro tenga

### 2.2.3 Múltiples Vistas Coordinadas

Independientemente del modo o técnica de visualización utilizada, el uso de *Múltiples Vistas Coordinadas (MVC)* junto con la posibilidad de interacción, está cada vez más presente. Esta técnica de visualización exploratoria permite al usuario exactamente eso, explorar los datos. Su premisa fundamental es que los usuarios entienden mejor los datos si pueden interactuar con ellos y verlos desde diferentes perspectivas al mismo tiempo; esto abre un gran abanico de posibilidades de análisis a la vez que eleva el grado de implicación y el nivel cognitivo del usuario.

Los usuarios quieren analizar complejos e intrincados conjuntos de datos; quieren explorar y encontrar patrones o comportamientos que no son fáciles de encontrar a simple vista. Este tipo de investigaciones requieren que el usuario considere muchos posibles escenarios y observe cómo se comportan cada una de las diferentes vistas. Estos complejos análisis requieren que las herramientas de exploración sean comprensivas e intuitivas.

## 2.3 Tecnologías de visualización

Actualmente las visualizaciones a través del navegador web se han hecho predominantes debido a su fácil acceso por cualquier sector de la población, independientemente del sistema operativo o dispositivo usado (smartphone, tablet, ordenador portátil o de escritorio). Esto ha sido posible gracias a la actual potencia de cálculo de cualquier dispositivo multimedia (un smartphone hoy en día tiene mayor potencia que un pc de escritorio de hace diez años). La base sobre la que se sustentan todas las visualizaciones web *dinámicas* es JavaScript. Este lenguaje de *scripts*<sup>10</sup> es de

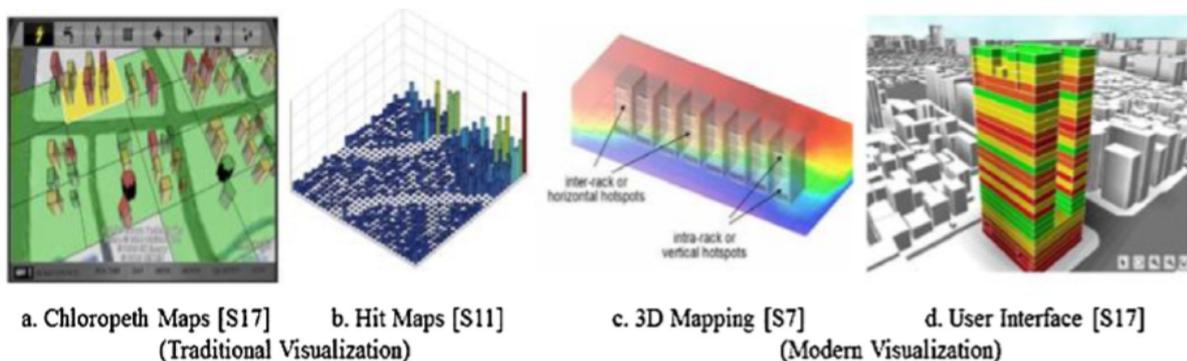


FIGURA 2.5. Tipos de visualizaciones 3D. Fuente - Design criteria for visualization of energy consumption: A systematic literature review [15].

el aspecto de una casa, o que una esfera parezca un balón de fútbol.

<sup>10</sup>Es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano; son casi siempre interpretados, pero no todo programa interpretado es considerado un script. El uso habitual de los scripts es realizar

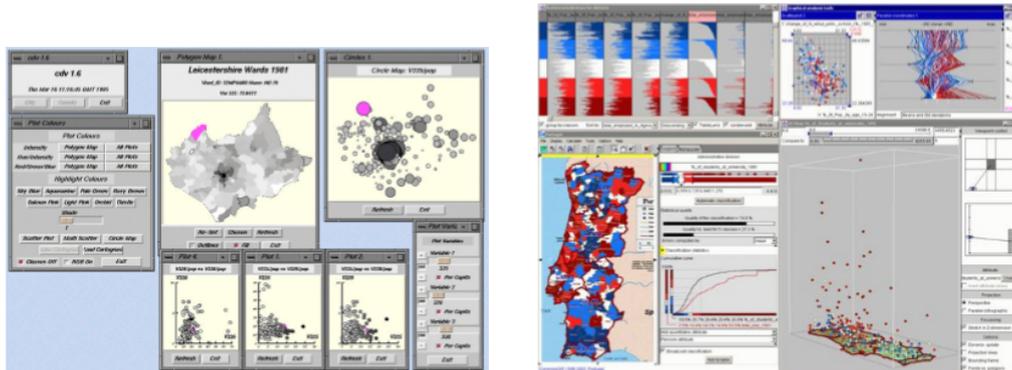


FIGURA 2.6. Capturas de aplicaciones que hace uso de múltiples vistas coordinadas (extraídas de "State of the Art: Coordinated & Multiple Views in Exploratory Visualization" [20]).

uso común en los navegadores y dispone, por ello, de multitud de librerías adaptadas a cualquier propósito; la más conocida en el campo de la visualización es D3.js, una librería sobre la que se sustenta el trabajo expuesto en esta memoria y sobre la que se hablará en profundidad en el siguiente capítulo. Derivadas de esta, se han desarrollado otras como son *nvd3.js* o *dc.js* (a un nivel superior), las cuales facilitan el trabajo al usuario al no tener que crear todo desde cero, y otra que trabaja en paralelo con esta, *crossfilter.js*, la cual es casi indispensable a la hora de trabajar con *múltiples vistas coordinadas* debido al extremadamente rápido uso de los filtros cruzados que ofrece.

Otras librerías en JavaScript que permiten crear y trabajar con gráficos vectoriales y objetos en 3D son *paper.js*, *rapahel.js*, *three.js* (totalmente orientado al 3D), *pixi.js*, *etc...*, aunque su uso para el análisis de datos, concretamente los de demanda energética, no es nada común.

Fuera de JavaScript, hay otras tecnologías como son *Processing*, lenguaje de programación y entorno de desarrollo integrado de código abierto basado, inicialmente, en Java. Es un lenguaje fácil de utilizar ya que fue creado para enseñar a programar a personas ajenas a este campo. Su target principal son los estudiantes, artistas, diseñadores e investigadores, los cuales hacen uso de él para prototipar de forma sencilla. Debido al, ya comentado, auge de JavaScript, se creó *Processing.js*, el cual traducía el código escrito en Java (lenguaje cada vez más en desuso) a JavaScript. Posteriormente, se desarrolló una librería para poder programar directamente en JavaScript, sin traducciones intermedias, esta librería se llama *P5.js*. La figura 2.7 muestra un ejemplo de visualización de consumos eléctricos desarrollada con *Processing*.

No pueden quedarse fuera las disponibles para otros lenguajes como son *Python* y sus librerías

---

diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario. Por este uso es frecuente que los shells sean a la vez intérpretes de este tipo de programas. Fuente - Wikipedia

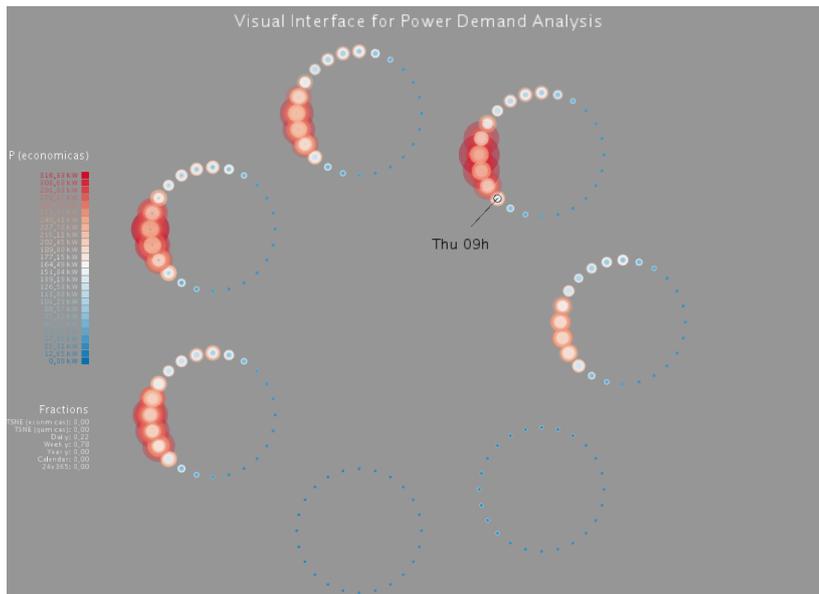


FIGURA 2.7. Visualización de consumos eléctricos desarrollada con Processing (creado por Díaz et al.[8]).

*matplotlib*, *pandas* y *mayavi2/VTK* (entre otras), *Gephi* (visualización de redes con miles de nodos), *Matlab* o *R*.

## 2.4 Encuadre del proyecto

Este trabajo se ubica dentro de lo que anteriormente se ha descrito como *visualizaciones 2D modernas*. Esto es así ya que, a pesar de usar gráficos 2D simples como son los histogramas (uno de ellos es un histograma bivariable en forma de scatter), permite interacción fluida, hace uso intensivo de *múltiples vistas coordinadas* y dispone de interface de usuario, en forma de botones, que habilitan diferentes tipos de consultas o búsquedas.

En cuanto a las tecnologías utilizadas, hay dos bien diferencias en función de la parte de proceso. La primera, el curado de los datos, ha usado la librería *pandas* de Python, mientras que para la propia creación de la herramienta de análisis se ha optado por las librerías *D3.js* en combinación con *crossfilter*, ambas de JavaScript, para hacerla compatible con todos los navegadores web sin falta de recurrir a *plugins*<sup>11</sup> externos.

<sup>11</sup>Es una aplicación que se relaciona con otra, como puede ser en este caso el navegador web, para agregarle una función nueva y, generalmente, específica.



## MÉTODOS Y TÉCNICAS

Para la realización de la aplicación se han utilizado distintos lenguajes de programación en función de la finalidad buscada y de la fase del proceso de creación. Además, en lo referente al aspecto visual, se han seguido ciertos criterios de diseño para conseguir una fácil legibilidad y usabilidad, independientemente de la cualificación del usuario relativa a al tema tratado.

### 3.1 Técnicas

Desde un principio se ha considerado que el soporte sobre el que se realizaría la visualización -interacción de la aplicación tendría que ser el navegador web. Esta elección se sustenta en varios puntos:

- Ausencia de instalación de cualquier tipo de software en el equipo receptor
- Independencia total del sistema operativo usado (Windows, MacOS, Linux, Android, etc...)
- Compatibilidad total con cualquier tipo de dispositivo, ya sea un pc, tablet o smartphone
- Derivado de lo anterior, la posibilidad de hacer uso de ella en cualquier lugar, ya sea en el trabajo, casa, calle, etc...

Debido a que la aplicación será ejecutada por el navegador del usuario, un procesado previo de los datos con el fin de reducir los más de medio millón de los que se dispone inicialmente, es de vital importancia. No siendo esta la única razón, sí que es de gran ayuda, ya que se desea

una alta fluidez en la interacción con la aplicación sea cual sea el dispositivo desde el que se esté usando, incluso teléfonos móviles.

Otras razones son el poder dar forma a los datos, curarlos<sup>1</sup> y agregar información correspondiente a cada una de las mediciones como puede ser la fecha y hora, universidad del que proceden, si ocurrieron en un día festivo, etc...

Todo este tratamiento de datos se hace en Python usando una librería para el análisis y tratamiento de estos llamada Pandas.

### 3.1.1 Tratamiento previo de datos: Python y Pandas

La elección de Python para esta fase del proyecto viene íntimamente ligada a la existencia de la librería *Pandas*. Siendo esta una razón de peso, no es la única, como se verá posteriormente una vez que se resuman algunas de sus características mas importantes.

"Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca el código legible"[29], es multiparadigma, de tipado dinámico y *multi-plataforma*. Posee una licencia de código abierto que es compatible con la *Licencia pública general de GNU* a partir de la versión 2.1.1. Todo esto hace que, actualmente, sea el lenguaje más extendido tal y como muestra la figura 3.1.

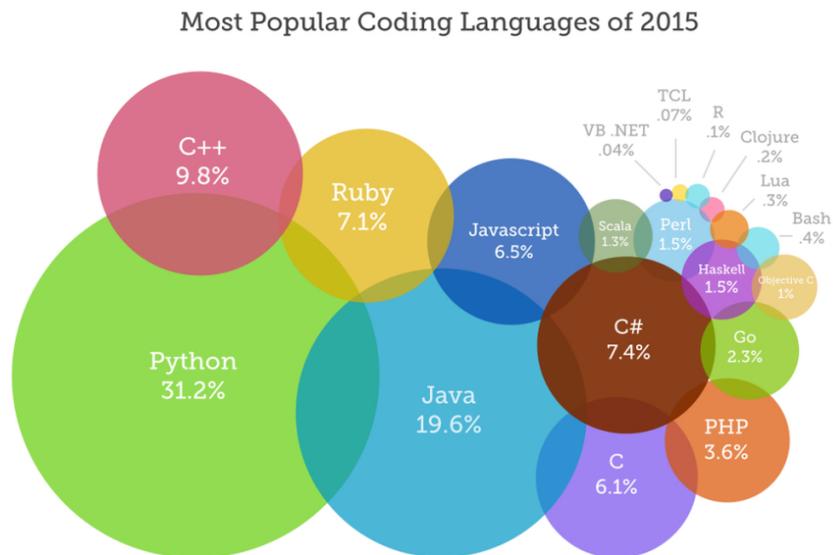


FIGURA 3.1. Porcentajes de uso de los lenguajes más usados en el 2015. Figura extraída de CodeEval [5].

<sup>1</sup>Forma coloquial de referirse al proceso de preparar los datos para su posterior manipulación ya sea eliminando incongruencias, rellenando valores vacíos, etc...

	a	b	c	d	e
2015-09-01 19:00:00	0.854972	1.810486	2.922159	1.331018	0.259328
2015-09-01 19:01:00	1.677018	-0.335902	0.411026	0.330194	1.867572
2015-09-01 19:02:00	0.134847	-0.404416	0.492147	-0.883389	0.204517
2015-09-01 19:03:00	-1.658516	0.375824	0.539066	0.515647	-0.417802
2015-09-01 19:04:00	-0.808565	-0.847851	-0.341983	-1.628785	0.383863
2015-09-01 19:05:00	-0.127820	-0.599479	-0.626599	0.227835	0.127312
2015-09-01 19:06:00	0.301348	0.110126	0.607690	0.786501	1.141093
2015-09-01 19:07:00	-0.466753	0.544194	-0.497510	-0.862704	-1.165716
2015-09-01 19:08:00	0.870780	0.061399	1.303483	0.249223	-1.142376
2015-09-01 19:09:00	-0.103090	-1.074768	0.010758	-0.408814	-1.998617

FIGURA 3.2. DataFrame de Pandas. La primera columna, la cual no tiene nombre o título, es el índice.

Esta gran implantación ha hecho posible el inmenso número de librerías con las que cuenta actualmente, abarcando, casi en su totalidad, todos los campos de la ciencia. En el caso de nuestro proyecto se usarán dos: *NumPy*, la cual proporciona la posibilidad de cálculos matriciales a un nivel similar a *MatLab*, haciéndose indispensable para que la otra librería *Pandas* pueda funcionar. Es esta última la que, de manera muy sencilla, nos permite limpiar y dar formato a los datos.

"Pandas contiene estructuras de datos y herramientas de manipulación de alto nivel diseñadas para hacer que el análisis de datos sea rápido y fácil"[13] . A estas estructuras se las llama *DataFrame* (*Serie* si trabajamos con datos unidimensionales ) y constan de un índice que identifica cada fila y de columnas que hacen lo propio con cada uno de los diferentes atributos. En la figura 3.2 se puede ver la forma de la estructura *DataFrame*.

### 3.1.2 El estándar Web (Html, CSS y JavaScript)

Un estándar es un conjunto de reglas normalizadas que debe cumplir un producto, servicio o proceso con el fin de establecer un mecanismo que permita conseguir plena compatibilidad, entre hardware y software, de todos los elementos que lo utilicen [23]. *World Wide Web Consortium (W3C)*, es el consorcio internacional encargado de producir las recomendaciones o estándares que aseguren el crecimiento a largo plazo de la Web. En la creación de estas recomendaciones participan mas de 400 organizaciones distribuidas por todo el mundo, el equipo de *W3C*, expertos internacionales o los propios usuarios.

Algunos de los estándares Web más conocidos son:



FIGURA 3.3. Jerarquía de DOM (Creado por I, JohnManuel y disponible bajo la licencia CC BY 2.5 vía Wikimedia Commons).

- HTML (*HyperText Markup Language*), sirve para definir la estructura de los documentos
- CSS (*Cascading Style Sheets*), permite asignar estilos para la representación de los documentos
- XML (*eXtensible Markup Language*), sirve de base para un gran número de tecnologías
- SVG (*Scalable Vector Graphics*), sirve para describir gráficos vectoriales bidimensionales en formato XML

Complementando lo anterior y, para poder trabajar con ello, está el *DOM (Document Object Model)*, que es una API<sup>2</sup> que representa un conjunto estándar de objetos para representar documentos (HTML, XHTML o XML), modelo sobre como pueden combinarse dichos objetos y una interfaz para acceder a ellos y manipularlos. Mediante el DOM, los programas pueden acceder al contenido, estructura y estilo de los documentos y manipularlos [27].

Para poder trabajar con el DOM de manera dinámica se tiene *JavaScript*, un lenguaje de scripting, multiplataforma y orientado a objetos, que es de uso común para la creación de páginas Web con contenidos dinámicos. Permite, mediante scripts, la interacción del usuario, el control del navegador Web y alterar el contenido del documento mostrado en este [10]. Se suele ejecutar del lado del *cliente*<sup>3</sup> aunque también puede trabajar del lado del *servidor* gracias a *Node.js*, un entorno de ejecución para el desarrollo de aplicaciones del lado del servidor.

<sup>2</sup> *Application Programming Interface*: es el conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción[25].

<sup>3</sup> La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. En este caso, los clientes son los propios usuarios desde sus equipos, ya sean PCs, tabletas o smartphones.

*JavaScript* es capaz de ser leído e interpretado por cualquier navegador Web, incluso los incorporados en los teléfonos móviles, ya que sigue el estándar *ECMAScript*[28] (actualmente aceptado como el estándar ISO 16262). Esto posibilita la casi total compatibilidad con cualquier dispositivo conectado a la red y, de ahí, su extendido uso.

### 3.1.3 D3.js y CrossFilter.js

De todas las librerías de JavaScript dedicadas a visualización, hay una que sobresale especialmente por encima de todas las demás, esta es D3 (Data-Driven Documents), pilar fundamental en el desarrollo de la aplicación Web aquí tratada. Según el artículo publicado por su creador, Michael Bostock, la definía como "... una novedosa forma de representación con un enfoque transparente para la visualización Web. En lugar de ocultar el escenario subyacente con una capa de abstracción específica del toolkit<sup>4</sup>, D3 permite la inspección y manipulación directa de una manera nativa: el estándar Document Object Model (DOM)"[4].

D3 es una librería que permite crear visualizaciones dinámicas e interactivas en los navegadores Web de una forma muy eficiente ya que trabaja directamente sobre el DOM. Hace uso de elementos y/o atributos HTML5, SVG y CSS, permitiendo crearlos, seleccionarlos o modificarlos de forma dinámica.

Una de las grandes ventajas de trabajar con los elementos de forma nativa<sup>5</sup> es la facilidad para la depuración. El método de *prueba y error* es una parte fundamental en el proceso de aprendizaje y desarrollo, por eso, el trabajar sin capas de abstracción, facilita al programador el poder visualizar los efectos de sus operaciones sin encontrarse errores como *falta de concordancia* en la capa, que violan el modelo mental del usuario.

*Data-Driven Documents (D3)* nace como evolución de *Protovis*, también del mismo creador, pero antes de estos hubo otros dos que merece la pena mencionar, estos son:

- Prefuse, toolkit realizada en 2005 en Java que necesitaba un plugin Java para renderizar en los navegadores.
- Flare, toolkit desarrollada en 2007, similar a Prefuse, que necesitaba un plugin Flash para el renderizado.

En 2009, basándose en la experiencia adquirida tras desarrollar y utilizar los dos toolkit anteriores, el profesor Jeff Heer, y los estudiantes Michael Bostock y Vadim Ogievetsky del Grupo de Visualización de la Universidad de Stanford, crean *Protovis*, una librería en *JavaScript* para generar gráficos SVG a partir de datos. Esta librería fue objeto de notable aceptación tanto

<sup>4</sup>Conjunto de herramientas de un software determinado

<sup>5</sup>sin capas de abstracción

por los profesionales de la Visualización de datos como por los académicos [26]. Las principales diferencias con su predecesor (*D3*) eran, aparte de las obvias en optimización, principalmente dos:

- Protovis sólo renderizaba en el SVG, mientras que D3 puede utilizar cualquier objeto DOM.
- Protovis proporcionaba una capa de abstracción simplificada para definir las propiedades visuales del documento mientras D3 trabaja directamente con CSS y las propiedades de los objetos DOM.

Actualmente hay alternativas como P5.js [17], la versión en JavaScript de Processing[18], o InfoVis[2], pero están lejos de la eficiencia y versatilidad de D3 debido a las limitaciones subyacentes de trabajar con capas de abstracción. Por contra, su curva de aprendizaje es menos pronunciada, posibilitando vistosas representaciones al poco tiempo de iniciarse en ellas.

Otro de los pilares sobre los que se sustenta la aplicación es *Crossfilter.js*, un rápido filtro multidimensional para vistas coordinadas [7]. Librería de gran utilidad para la exploración de grandes conjuntos de datos multivariable en el navegador Web y cuya principal ventaja es su extrema rapidez (<30ms) [7] en la interacción con las vistas coordinadas, incluso con *datasets*<sup>6</sup> de mas de un millón de de muestras. Para conseguir esto sólo trabaja con una dimensión cada vez que se lleva a cabo una interacción por parte del usuario, lo cual parece suficiente ya que no se precisa de más en la gran mayoría de situaciones. Básicamente, su modo de trabajo es aplicar

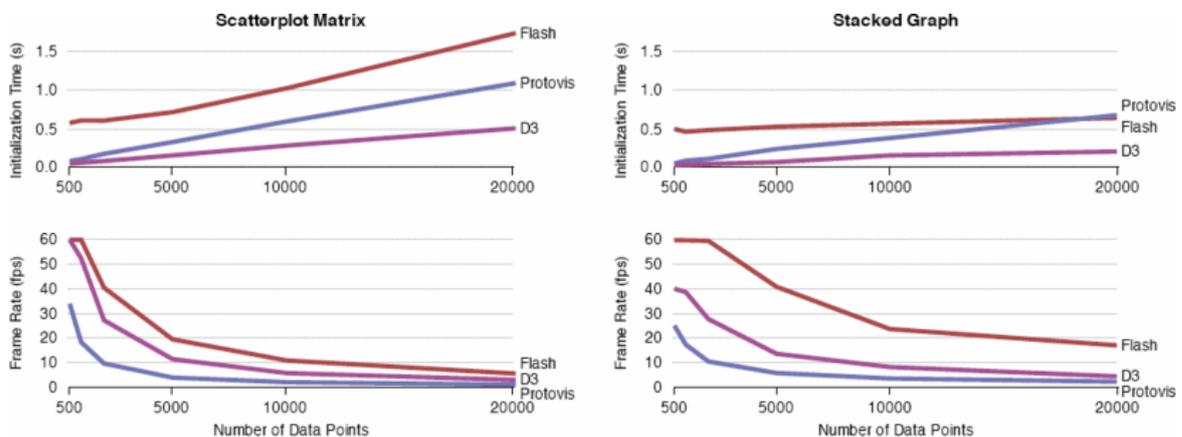


FIGURA 3.4. Comparativa de rendimiento D3 frente a su precursor Protovis y el ya en desuso plugin Flash. Test realizado en un MacBook Pro dual core 2.66GHz, 8GB de RAM corriendo MacOS 10.6.7 sobre Chrome 11. Figura extraída del artículo Data-Driven Documents (D3) de M. Bostock[4].

<sup>6</sup>Conjuntos de datos.

filtros incrementales o decrementales a los valores de partida, lo cual es más rápido que empezar desde cero cada vez en cada interacción.

Todo en el ámbito de *Crossfilter* está dentro del espacio de nombres *crossfilter*, el cual es, además el constructor<sup>7</sup>. Sobre este objeto se crearán tantas dimensiones como sean las necesarias y será, desde éstas, donde se realizará todo el trabajo posterior de filtrado. Es conveniente indicar que el mejor rendimiento se obtiene cuando el número de dimensiones creadas no excede el de ocho.

*Crossfilter* da lo mejor de sí cuando trabaja junto a *D3*, tal y como se puede ver en el ejemplo mostrado en la propia web de la librería [7]. Es, además, tanto una gran muestra de su potencial, como una fuente de inspiración y gran punto de partida para la creación de nuevas aplicaciones de visualización.

A pesar de que la librería de nivel superior *dc.js* contiene tanto a *d3.js* como a *crossfilter.js* y facilita enormemente el trabajo al tener funciones que crean, de manera sencilla, gráficas coordinadas con buena estética, se ha desechado su uso para poder trabajar con las dos librerías de forma nativa y mejorar con ello, la fluidez de la herramienta.

## 3.2 Criterios de diseño

A la hora de crear un correcto entorno gráfico que pueda informar, a la vez que ayudar al usuario final, a sacar sus propias conclusiones sobre el consumo de energía de un determinado complejo, es de vital importancia atenerse a unos principios básicos de diseño. Estos se pueden englobar en dos grupos o áreas principales bien diferenciadas, *criterios funcionales* y *criterios no funcionales*[19]. Los criterios funcionales comprenden las siguientes categorías:

- Información mostrada en la visualización
- Modos de visualización
- Técnicas de visualización

Mientras que los no funcionales son los que tienen en cuenta las consideraciones en cuanto a hardware y software apropiadas.

En esta sección se presentarán estas dos categorías que, juntas, forman el *Criterio de diseño para visualización del consumo de energía*, resumido en la figura 3.5.

---

<sup>7</sup>En el ámbito de la programación orientada a objetos, los constructores son funciones miembro especiales que sirven para inicializar un objeto de una determinada clase al mismo tiempo que se declara.

Criteria	Categories	Concepts
Design criteria for visualization		
Functional criteria	Information displayed in the visualization	Information suitable for end-users Information suitable for energy managers Information suitable for both residents and energy managers
	Techniques in visualization	2D visualization 3D visualization
	Modes of visualization	Modes based on time, scale and information
Non-functional criteria	Non-functional software considerations	Flexibility, scalability, etc. (refer <a href="#">Table 4</a> for full list)
	Non-functional hardware considerations	Size of display, placement of display, viewing distance, etc.

FIGURA 3.5. Sumario de los criterios de diseño. Figura extraída del artículo *Desing criteria for visualization of energy consumption: A systematic literature review* [15].

### 3.2.1 Criterios funcionales

#### 3.2.1.1 Información mostrada en la visualización

Cuando se desea crear una representación visual de un conjunto de datos, en particular la energía consumida, se ha de tener en cuenta hacia quien irá dirigida, de forma que se pueda mostrar todo de la forma más conveniente para el usuario. En este punto, tenemos dos grupos claramente diferenciados, los *usuarios finales*, consumidores de energía en el entorno residencial, y los *gestores energéticos*, profesionales entrenados que entienden perfectamente la información que se pueda representar de forma muy eficiente y que monitorizan el consumo de energía en universidades, edificios públicos, empresas, etc.

En ambos casos la meta es ayudar al usuario a tomar decisiones concernientes al consumo de energía y, en consecuencia, a una mejor conservación de esta y menor gasto.

**Información apropiada para usuarios finales:** Constituyen el mayor grupo y, por ende, el más importante, ya que en sus manos está el poder reducir el consumo energético en los hogares o, en todo caso, hacer que se haga de una forma más eficiente. Este grupo es muy heterogéneo debido a las grandes diferencias que se pueden encontrar de un usuario a otro en función de su nivel educativo, habilidad analítica, afinidad con la tecnología, etc.

**Información apropiada para gestores energéticos:** Además de los espacios residenciales, se consume una gran cantidad de energía en los edificios públicos como pueden ser las universidades. Sin embargo, a diferencia del hogar, estos espacios son monitorizados por personas cualificadas encargadas de tener bajo control el consumo de energía. Estos profesionales pueden entender y manejar visualizaciones más complejas si fuera necesario.

**Información apropiada para ambos, usuarios finales y gestores energéticos:** Esta es

la meta a la que aspira la aplicación de visualización creada en este proyecto, ser una herramienta que pueda ser utilizada por la mayor cantidad posible de personas sin importar su nivel de formación. Que permita a estas entender y sacar conclusiones de los patrones de consumo observados, con el fin de conseguir la mayor *Eficiencia Energética*

### 3.2.1.2 *Modos de visualización*

El papel fundamental de la visualización es el de proporcionar a los usuarios un *feedback* sobre el consumo energético. Los modos de visualización se pueden clasificar en base a su granularidad con respecto a tiempo, escala o tamaño, tipo de información mostrada y técnicas (descritas en la siguiente sección). Ejemplos de estos son los siguientes:

- Los basados en la información del consumo de energía pueden ser de la forma de kilovatio-hora (kWh), kilojulio (kJ), cantidad de CO2 emitida, etc.
- Los basados en periodos de tiempo pueden ser por horas, días, meses, etc.
- Los basados en escala pueden indicar el consumo de cada una de las habitaciones o departamentos o el total de todos ellos, etc.

La visualización creada tiene en cuenta todos y cada uno de ellos ya que muestra kWh, factor de potencia o euros; se puede ver por horas, días de la semana, meses, año y festivos y de forma totalizada o individual para cada edificio<sup>8</sup>. La mayor parte de la visualización ha sido representada en forma de diagramas de barras debido a su gran implantación y, por ello, fácil interpretación.

### 3.2.1.3 *Técnicas de visualización en el consumo de energía*

Como ya se ha visto en el capítulo previo, existen dos bien diferenciadas, la visualización 2D y la visualización 3D, ambas pudiendo ser abordadas desde un enfoque tradicional o moderno.

En el caso que nos ocupa se ha optado por la representación 2D siendo utilizado, en la mayoría de los casos, con gráficos simples como histogramas para , como ya se ha comentado, favorecer siempre la perfecta comprensión de la información mostrada, sin importar los conocimientos o formación del usuario. El enfoque, aun así, es en todo momento moderno al posibilitar la interacción y ofrecer una sencilla interface de control.

---

<sup>8</sup>En este caso cada edificio indica la totalidad de una universidad, pero podrían ser habitaciones de un hogar, diferentes departamentos de una empresa, etc.

### 3.2.2 Criterios no funcionales

Son aquellos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de este como la fiabilidad, respuesta en el tiempo, capacidad de almacenamiento, interface del sistema, etc. Se dividen principalmente en dos grupos, las *consideraciones de hardware* y las *consideraciones software*.

#### 3.2.2.1 Consideraciones de hardware

A este grupo pertenecen consideraciones como el tipo<sup>9</sup>, tamaño y resolución de la pantalla que se va a usar para la visualización, la distancia de visionado (una mayor distancia de visionado debe repercutir en un mayor display que proporcione representaciones del tamaño adecuado) y la potencia mínima requerida para un correcto desempeño de la aplicación que permita una interacción fluida.

Aquí se ha optado por unas consideraciones bastante modestas para lo acostumbrado en los tiempos que corren. Un portátil cualquiera puede brindar una excelente experiencia de uso al haberse desarrollado con ellos en mente como principales dispositivos. Aún así, debido al auge de las tabletas y los smartphones, puede ser ejecutada también en estos, consiguiéndose resultados bastante satisfactorios. Para ello, el dispositivo debe contar con, al menos, cuatro núcleos a más de 1.5Ghz. Otro aspecto a tener en cuenta para la correcta visualización, es que el formato de la pantalla sea de 3/2, ya que es este el del monitor que se ha usado a lo largo de todo el desarrollo de la aplicación.

#### 3.2.2.2 Consideraciones de software

La figura 3.6 muestra los seis puntos básicos que debe intentar seguir cualquier producto software y que, en el caso de este proyecto, se ha conseguido casi en su totalidad.

- *Portabilidad*, habilidad de la aplicación para trabajar en múltiples plataformas y dispositivos.
- *Accesibilidad*, referido a la facilidad con la cual los usuarios pueden acceder a ella online.
- *Flexibilidad*, la cual hace mención a la habilidad para cambiar entre diferentes modos de visualización o datos visualizados.
- *Comprensibilidad*, referida a la facilidad con la que los usuarios interpretan lo que se les muestra y aprenden a usar todas sus posibilidades.
- *Escalabilidad*, la habilidad de la aplicación para acoger futura información en la visualización.

---

<sup>9</sup>Puede ser monocromo o a color, táctil, tener una relación de aspecto 3/2, 16/9, etc.

- *Extensibilidad*, facilidad para los desarrolladores de extender la aplicación de una forma sencilla y sin modificar prácticamente el código existente.

Non-functional software criteria	Criteria description
Portability	Ability to work on multiple platforms and devices
Accessibility	Ability to download the feedback from online sources for offline processing
Flexibility	Ability to find, download and use the visualization application online, such as on websites or social networking sites
Understandability	Ability to toggle between different modes of visualization
Scalability	The ease in recognition of information displayed in the visualization by the human mind
Extensibility	The ability of the application to scale in order to accommodate future information in the visualization
	The ease for the developers to extend the application

FIGURA 3.6. Consideraciones software no funcionales y sus descripciones. Figura extraída del artículo *Desing criteria for visualization of energy consumption: A systematic literature review* [15].

## 3.3 La aplicación: plantilla html, archivos y funciones JavaScript

### 3.3.1 Resumen del proceso previo realizado Python

Debido a las facilidades que nos brinda Python y Pandas, descritas en el apartado 3.1.1, el proceso seguido es muy sencillo. Se parte de tres documentos Excel, dos con los datos de consumo de las dos universidades objeto de estudio (Química y Económicas), y el último con los precios/hora de cada una de las 24 horas del día.

Todos estos archivos son importados con la siguiente instrucción:

```
pd.read_excel("\ruta_del_archivo/nombre_del_archivo.xls");
```

donde `pd` hace mención a la librería *pandas*. El archivo importado, de tipo `DataFrame`, se almacena en una variable ( la llamaremos `Dataframe` de aquí en adelante). Sobre esta, podremos seleccionar las filas y columnas que nos interesen para el estudio ya que, en la mayoría de los casos, tendremos varias en blanco o con valores no deseables. Esto se hace con el siguiente comando:

```
Dataframe = Dataframe.iloc[filas ,columnas];
```

Una vez que tenemos sólo los datos que queremos, les ponemos cabecera a cada una de las columnas así como un índice a cada una de las filas.

```
Dataframe.index = Dataframe[ 'Indice '];
Dataframe.columns = [ 'Columna1', 'Columna2', 'Columna3', ...];
```

Para reducir el número de muestras de más de 35.000 por Universidad, los datos fueron tomados cada 15 minutos, a 8.760 que son las horas de un año, se hace uso de una instrucción muy potente de Pandas llamada *resample*. El valor de las horas será la media de las 30 muestras tomadas en esa hora.

```
Dataframe = Dataframe.resample('1h', how='mean');
```

En estos momentos ya estamos en disposición de empezar a agregar todos los datos que necesitemos en forma de nuevas columnas; a estas se las conoce como "*helper columns*" y permiten establecer modos de análisis complementarios (por horas, días, semanas,...) que aportan visiones diferentes de los datos. En este caso han sido la hora, el día de la semana, el mes, el edificio o universidad a la que pertenecen y si es festivo o no.

Por otra parte, uno de los problemas que a menudo se suele encontrar a la hora de trabajar con datos es, paradójicamente, la falta de ellos. Esto, además, se acrecienta ya que se dan en posiciones completamente aleatorias. A estos datos ausentes se les conoce como *NANs* (*Not A Number*) y pueden ocasionar un gran perjuicio, no sólo a la hora de visualizarlos, sino durante la propia ejecución del programa. La instrucción *fillna* de Pandas hace sencilla la resolución de este problema. Esta sustituye los *NANs* que encuentre por valores de otra matriz o *DataFrame* que se le indique. En nuestro caso, ese *DataFrame* con los datos de sustitución será uno que creemos previamente y que tendrá los valores medios de las potencias a la hora y día de la semana indicados. Esta elección es una buena aproximación para rellenar datos vacíos ya que no sólo tiene en cuenta la hora, sino también el día de la semana.

Para obtener ese *DataFrame*, tenemos otra instrucción de Pandas que es *groupby*,

```
DataframeSustitucionNANs = Dataframe.groupby('DiaSemana', 'Hora').mean();
```

aplicándolo junto a *fillna* obtenemos un *DataFrame* libre de *NANs*

```
Dataframe.Columna = Dataframe.columna.fillna(DataframeSustitucionNANs);
```

Se unen los *DataFrames* de cada Universidad ...

```
DataframeConjunto = pd.concat(DataframeQuimicas, DataframeEconomicas);
```

... y exporta en formato *json* ya que es el estándar en *JavaScript* y facilitará mucho las cosas posteriormente.

```
DataframeConjunto.to_json('Nombre_del_json', orient='records', date_format='iso');
```

### 3.3.2 Plantilla de la Aplicación: HTML y CSS

La estructura sobre la que se ha creado la aplicación, la cual se puede ver de manera muy resumida en la figura 3.7, consta de catorce bloques o div, así como de un script encargado de automatizar el proceso de creación de contenido sobre ellos. La elección de crear estos div es debida a que, desde la hoja de estilos CSS se puede asignar fácilmente el tamaño, formato o estilo deseado, sin tener que modificar el código html. Esto es sumamente útil, al poder adaptar la aplicación a otros formatos de pantalla como pueden ser los 16:9 de las televisiones o los smartphones, sin más que cambiar un par de parámetros en la hoja de estilos.

La distribución de estos DIV es la siguiente:

- Uno para alojar a título
- Tres para sendos bloques de texto encargados de informar al usuario sobre que muestra la aplicación y como se usa. Los nombres de estos bloques son *Problem description*, *Interaction mechanisms* y *Basic usage*.



```

1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml">
3
4 <head>...</head>
18
19 <body>
20 <div id="title">...</div>
23
24 <div id="description" class="des_mec_usa">...</div>
33
34 <div id="mechanisms" class="des_mec_usa">...</div>
47
48 <div id="usage" class="des_mec_usa">...</div>
59
60 <div id="Div_scatter" class="Div_General_Charts">...</div>
64
65 <div id="Div_activa" class="Div_General_Charts">...</div>
69
70 <div id="Div_cosPhi" class="Div_General_Charts">...</div>
74
75 <div id="Div_hora" class="Div_General_Charts">...</div>
79
80 <div id="Div_dia" class="Div_General_Charts">...</div>
84
85 <div id="Div_mes" class="Div_General_Charts">...</div>
89
90 <div id="Div_holiday" class="Div_General_Charts">...</div>
94
95 <div id="Div_edificio" class="Div_General_Charts">...</div>
99
100 <div class="margen"></div>
101
102 <div id="controls"></div>
103
104 <div id="tabla"></div>
105
106 <div class="margen" id="margen2"></div>
107
108 <footer>
109 <script>...</script>
146
147 </body>

```

FIGURA 3.7. Captura sintetizada de la plantilla HTML del proyecto.

- Ocho para las gráficas, siendo siete de estos los encargados de contener los histogramas al uso y, el otro, el que hará lo propio con el autodenominado *histograma 2D*, el cual ocupa todo el ancho de la pantalla.
- Dos encargados de crear márgenes. Uno entre los histogramas y la tabla (para un fácil rediseño desde la hoja de estilos CCS) y otro al final de la página para el correcto funcionamiento de las funciones de scroll<sup>10</sup>.

En la figura 3.8 se muestran capturas de las diferentes partes de la aplicación.

### 3.3.3 Archivos utilizados

Para la realización de esta aplicación se han utilizado una serie de archivos que se detallan a continuación:

- *HtmlCrossfilter.html*, es el que el navegador primero ejecutará y, por lo tanto, el que tiene que incluir la estructura de DIVs<sup>11</sup> de la aplicación, así como las llamadas a las funciones encargadas de crear todo el contenido que el usuario verá.
- *StyleSheet.css*, es la hoja de estilos de la aplicación Web; en ella se describe el tamaño de cada DIV así como los colores y demás propiedades de todos los objetos que serán utilizados.
- *data\_Univ.json*, es la estructura que contiene los datos que usaremos para su visualización y análisis. El formato .json (JavaScript Object Notation) es el idóneo para trabajar en JavaScript ya que es nativo de ese lenguaje y es leído e interpretado sin dificultad.
- *FuncionesTrabajoMaster.js*, es la librería que, como su propio nombre indica, contiene todas las funciones empleadas en la aplicación, ya sean para crear el contenido o para realizar únicamente cálculos. En el punto 3.3.4 se detallan todas y cada una de las funciones.
- *colorbrewer.js*, librería que contiene los patrones de color basados en el trabajo de Cynthia Brewer<sup>12</sup>, muy útiles en ciertas visualizaciones.
- *d3.slider.js* y *d3.slider.css*, librería junto con su correspondiente hoja de estilos, encargada de crear los *sliders* al usar la opción de modificar tarifa.
- *left-handle.png* y *right-handle.png*, imágenes utilizadas para representar los extremos de la selección realizada sobre cualquiera de los histogramas.

La figura 3.9 muestra una captura de todos ellos dentro del entorno de desarrollo usado.

---

<sup>10</sup>Desplazamiento a un lugar determinado de la página al tocar un botón.

<sup>11</sup>Cada uno de los bloques de contenido de una página web.

<sup>12</sup>Cartógrafa reconocida por su trabajo sobre el uso de los colores en representaciones cartográficas, haciendo hincapié en que estos sean fácilmente legibles y discernibles unos de otros.

### 3.3. LA APLICACIÓN: PLANTILLA HTML, ARCHIVOS Y FUNCIONES JAVASCRIPT

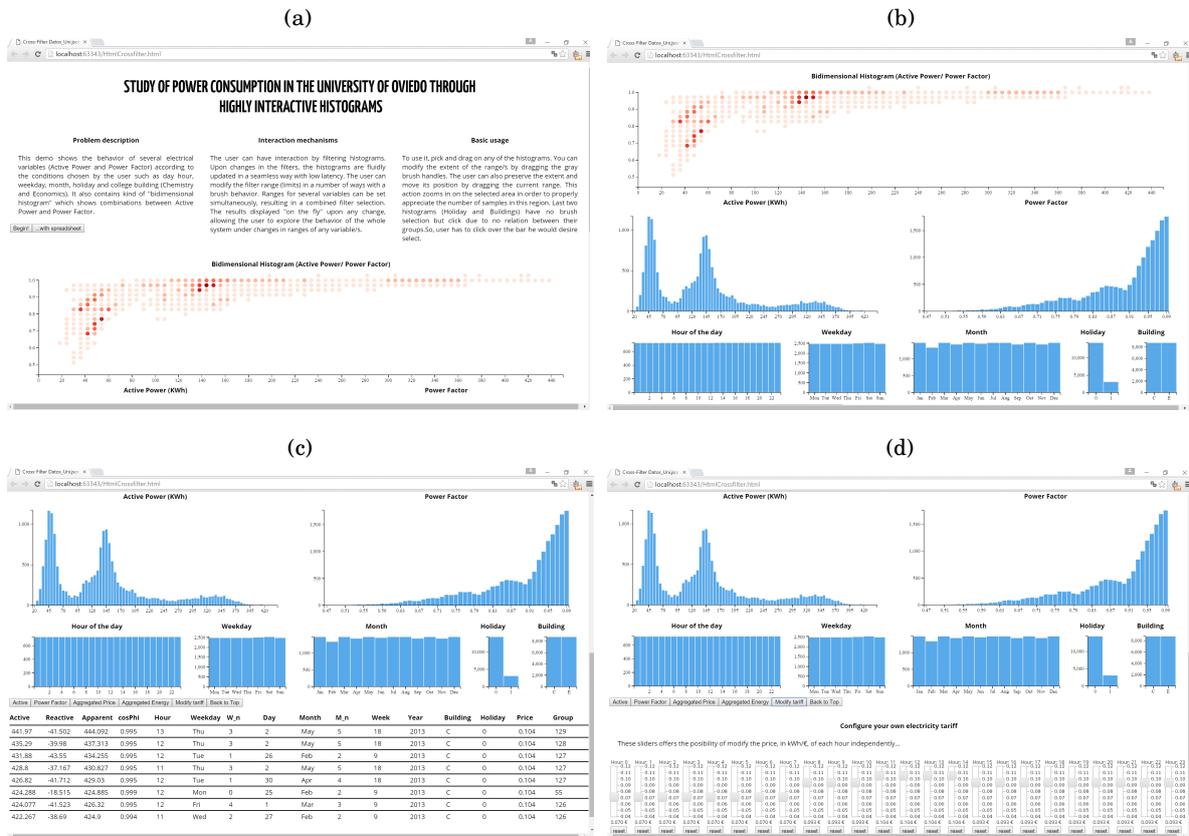


FIGURA 3.8. Capturas de las diferentes partes de la aplicación. En ellas se puede observar la metodología de diseño empleada la cual sigue los principios fundamentales de diseño analítico de E. Tufte [22] (a) Al cargar la web se puede ver el título, tres párrafos que describen la aplicación y su uso, y el histograma 2D. (b) Al pulsar el botón "begin!" (se ve en la figura a) se realiza un scroll para hacer que sólo se muestren en pantalla los histogramas, con el fin de poder analizar, sin distracciones los datos. (c) Similar a lo anterior pero tras pulsar "...with spreadsheet". En este caso el scroll baja hasta mostrar completamente la tabla, con lo que desaparece de la ventana el histograma 2D. (d) Desde la misma vista de la imagen anterior se accede a los sliders de modificación del precio/hora al pulsar el botón "Modify tariff" (es el segundo botón por la derecha que se puede ver en la propia imagen).

#### 3.3.4 Funciones JavaScript y nociones básicas de D3.js

En esta sección se profundizará en cada una de las funciones creadas para este proyecto, describiendo su funcionamiento y opciones que permiten, en función de los diferentes usos que se pueda requerir de ellas. Todas han sido escritas haciendo uso de D3 debido a sus potentes métodos de selección, creación y modificación de objetos y propiedades del DOM. Al final de esta sección, se podrán ver todas, de manera resumida, en la tabla 3.2.

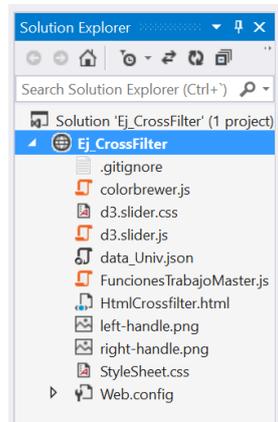


FIGURA 3.9. Captura del *Explorador de Soluciones* del Visual Studio 2013 que muestra los archivos utilizados en el proyecto

- `Crear_Dimensiones(DataArray)`: Esta función recibe como argumento el array de datos y se encarga de automatizar el proceso de creación del objeto crossfilter así como las dimensiones necesarias para llevar a cabo la aplicación. Antes de crear todos los elementos crossfilter, se llama a las funciones *Crear\_Grupos* y *agruparCombinaciones*, las cuales crean, en el `DataArray`, una columna adicional con el grupo al que pertenece cada combinación posible de kWh y factor de potencia medidos. La dimensión que se cree a partir de esas combinaciones, será la usada por la función *Crear\_Histo2D*.
- `Crear_Histogramas()`: No recibe ningún argumento y se encarga de automatizar el proceso de creación de todos los histogramas. Para ello, llama siete veces a la función *Crear\_Histo*, con los parámetros correspondientes para cada uno de ellos.
- `EliminarHistos()`: No recibe ningún tipo de argumento y borra todos los histogramas mostrados en pantalla. Para ello elimina el contenido del SVG de cada uno de los DIV destinados a las gráficas.
- `Crear_Histo2D(dimension, grupo, targetDiv)`: Esta función crea un scatter, con una sola dimensión de datos, en la que el color de cada círculo corresponde al número de muestras perteneciente a ese grupo. Para ello, se emplea el patrón de color Brewer<sup>13</sup> del rojo, siendo más oscuro a mayor número de coincidencias. Se ha llamado histograma 2D ya que, en realidad, no es más que un histograma representado como scatter, cuya variable es la combinación de dos de ellas. Recibe como argumentos la dimensión y el grupo de la columna agregada creada por la función *Crear\_Dimensiones* (la que contiene los grupos a los que pertenece cada muestra medida). Además, recibe también como argumento el

<sup>13</sup>Cartógrafa reconocida por su trabajo sobre el uso de los colores en representaciones cartográficas, haciendo hincapié en que estos sean fácilmente legibles y discernibles unos de otros.

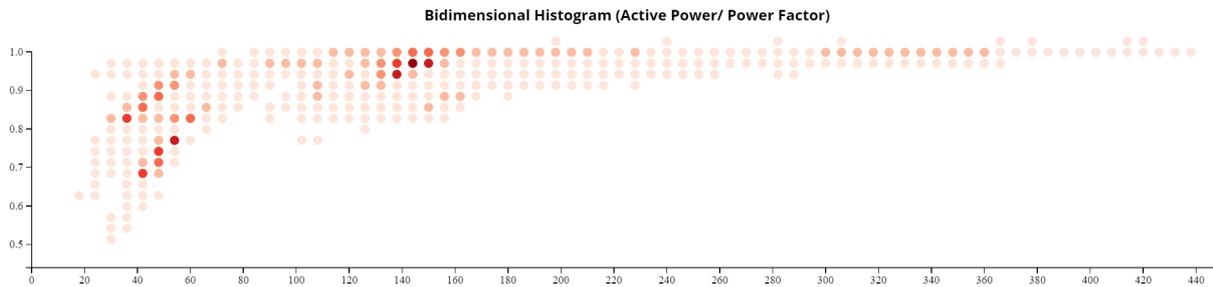


FIGURA 3.10. Aspecto de la gráfica creada con la función `Crear_Histo2D`. Captura extraída de la propia aplicación.

identificador *"id"*, del SVG en el que se ha de dibujar el gráfico. La figura 3.10 muestra el aspecto del gráfico creado.

- `Crear_Histo(dimension, grupo, targetDiv, ordinal, abajo, gruposConReduce, usarReduce)`: Función que permite crear un histograma común de barras cuyo eje horizontal muestra el valor de los grupos de una determinada dimensión y el eje vertical el número de muestras de ese determinado grupo. Son histogramas sobre los que el usuario puede interactuar seleccionando parte de los datos mostrados. El resultado de esta selección o filtrado se aplicará, instantáneamente, al resto de los histogramas dibujados siguiendo el principio de *vistas múltiples*. Recibe como argumentos la dimensión y grupo asociado que se quiere representar, el id del SVG de destino, la indicación de si los grupos son ordinales o no (si siguen un orden), la posición en la plantilla HTML (arriba o abajo)<sup>14</sup>, indicación de si el grupo ha sido creado usando la función *reduce* de *crossfilter*<sup>15</sup> y, por último, y en caso de que se haya usado *reduce*, si queremos usar los valores agregados de esta o no. Como se puede ver, es una función con múltiples opciones de configuración. La figura 3.11 muestra el aspecto de estos histogramas.
- `update()`: Esta función se encarga de redibujar todas los histogramas cada vez que el usuario interactúa con la aplicación, ya sea haciendo selecciones sobre los histogramas, modificando el tipo de datos a ser mostrados (al pulsar el botón correspondiente) o cambiando el precio de las tarifas horarias mediante los sliders.
- `Crear_Spreadsheet(dimension, targetDiv, max)`: Crea una tabla con todos los atributos de las 8 muestras de mayor energía consumida de la muestra seleccionada ese ese momento determinado. Por defecto se crea así, pero si el parámetro *"max"* se pone a cero,

<sup>14</sup>Los histogramas dibujados abajo en la plantilla tienen un menor tamaño en relación con los que van arriba.

<sup>15</sup>La función *reduce* calcula una serie de valores agregados en los grupos, por lo que la forma de acceder a los datos es distinta.

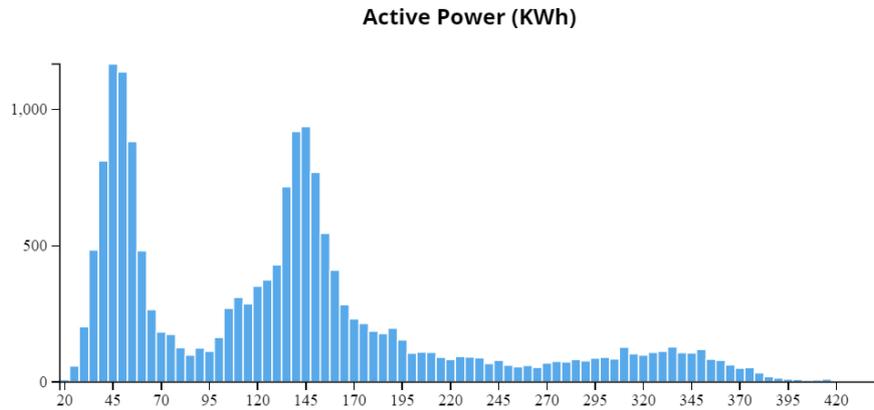


FIGURA 3.11. Aspecto de la gráfica creada con la función `Crear_Histo`. Captura extraída de la propia aplicación.

lo haría con las 8 muestras de menor energía. También recibe como argumento el SVG de destino donde será dibujada. La figura 3.12 muestra el aspecto de esta tabla.

- `Crear_controles()`: Crea todos los botones de la aplicación con sus respectivas funciones asociadas. Se pueden ver algunos de ellos en la figura 3.12.
- `Crear_valoresMax(target_Div)`: Recibe como argumento el *id* del DIV de destino, y muestra en él los valores máximos (más altos) de ciertos histogramas cuando el usuario pulsa los botones de "Aggregate Price" o "Aggregate Energy".
- `modifytariff()`: Crea los sliders de modificación tarifaria o la tabla con los 8 mayores consumos en función de lo que ya estuviera previamente en pantalla.
- `crear_slider(target_Div, group)`: Crea tantos sliders como número de elementos tenga el grupo que se pasa como argumento a la función, y los distribuye de forma que ocupen

Active	Reactive	Apparent	cosPhi	Hour	Weekday	W_n	Day	Month	M_n	Week	Year	Building	Holiday	Price	Group
441.97	-41.502	444.092	0.995	13	Thu	3	2	May	5	18	2013	C	0	0.104	129
435.29	-39.98	437.313	0.995	12	Thu	3	2	May	5	18	2013	C	0	0.104	128
431.88	-43.55	434.255	0.995	12	Tue	1	26	Feb	2	9	2013	C	0	0.104	127
428.8	-37.167	430.827	0.995	11	Thu	3	2	May	5	18	2013	C	0	0.104	127
426.82	-41.712	429.03	0.995	12	Tue	1	30	Apr	4	18	2013	C	0	0.104	127
424.288	-18.515	424.885	0.999	12	Mon	0	25	Feb	2	9	2013	C	0	0.104	55
424.077	-41.523	426.32	0.995	12	Fri	4	1	Mar	3	9	2013	C	0	0.104	126
422.267	-38.69	424.9	0.994	11	Wed	2	27	Feb	2	9	2013	C	0	0.104	126

FIGURA 3.12. Aspecto de la tabla creada con la función `Crear_Spreadsheet`. Captura extraída de la propia aplicación.

todo el ancho de la página. En el caso de la aplicación se crean 24, correspondientes a los 24 elementos del grupo de horas del día. En la figura 3.13 se puede ver una captura sacada de la propia aplicación.

- `slideFunc(evt, value)`: Esta función es llamada cada vez que el usuario desliza un slider o hace clic en cualquier punto de este. Se encarga de modificar los precios de la columna correspondiente de los datos originales y, a continuación, llamar a las funciones *EliminarHistos*, *Crear\_Dimensiones*, *Crear\_Histogramas* y *update* para la creación de nuevos grupos con sus correspondientes, y actualizados, valores de *reduce* y su posterior representación.
- `resetPrice()`: Devuelve el correspondiente slider al valor de precio inicial en caso de que hubiera sido modificado por parte del usuario. Llama a las funciones *EliminarHistos*, *Crear\_Dimensiones*, *Crear\_Histogramas* y *update* para devolver a los grupos los valores de partida y, a continuación, representarlos.
- `ordenarActiva()`, `ordenarCosPhi()`, `ordenarHora()`: Son funciones vinculadas a la pulsación de su respectivo botón y se encargan de ordenar la tabla en función de ese atributo. Tiene en cuenta el estado actual, por lo que cada vez que se pulse pasa de mayor a menor y viceversa. Como nota, el botón de ordenación por hora fue eliminado de la versión final.
- `CrearGrupos(matrizGeneral)`: Agrupa los valores de 2 de los atributos cualesquiera de la matriz de datos original. La resolución de cada grupo es diferente ya que los parámetros han sido ajustados para la aplicación que nos ocupa.
- `agruparCombinaciones(grupo1, grupo2, matrizGeneral)`: Toma los 2 grupos creados que la anterior función (*CrearGrupos*) y crea un nuevo atributo en la matriz original que es el indicador del conjunto al que pertenece esa combinación de valores de ambos grupos. Ej.si 2 muestras tienen el mismo grupo de activa y de cos phi, el nuevo atributo que indica el conjunto al que pertenecen, será el mismo.

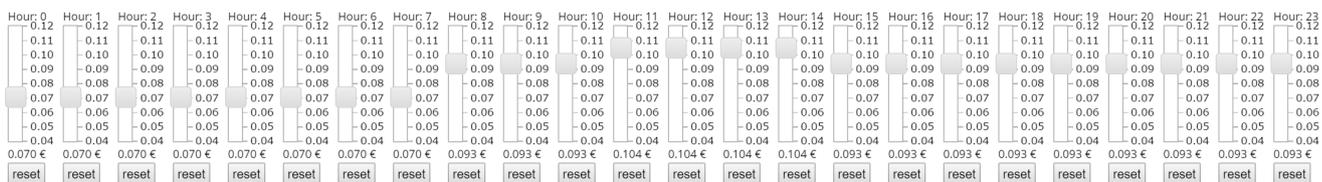


FIGURA 3.13. Aspecto de los sliders creados con la función `crear_slider`. Captura extraída de la propia aplicación.

- `agregadosPrice(grupos)`, `agregadosEnergy(grupos)`: Funciones que se encargan de modificar los títulos de los histogramas cuando estos pasan a mostrar costes en € o energía en kWh, tras pulsar los botones "*Aggregated Price*" o "*Aggregated Energy*".
- `reduceAdd(p, v)`, `reduceRemove(p, v)`, `reduceInit(p, v)`: Son las 3 funciones indispensables para poder crear grupos usando el potente método *reduce*. Este método forma parte de la librería de *Crossfilter.js* y permite realizar una serie de cálculos acumulativos sobre todos los elementos de un mismo grupo. Estos cálculos son realizados en el momento que se crean los grupos, ejecutando la función `reduceAdd` con cada uno de los registros e incrementando el contador de una serie de variables. La función `reduceRemove` hace exactamente lo contrario ya que, al reducir el número de elementos de los grupos (al hacer un brush sobre los histogramas), decrementa esos contadores. `reduceInit` simplemente inicializa los valores. Un ejemplo simplificado de la función `reduceAdd` sería el siguiente:

```
function reduceAdd(p, v) {
  ++p.count;
  p.hourPrice = v.Price;
  p.totalkWhGroup += v.Active;
  p.price += v.Active;

  if (p.count != 0) {
    p.avg = p.price / p.count;
    p.avgEnergy = p.totalkWhGroup / p.count;
    if (p.count > 24) {
      p.avgWeekday = p.price / Math.floor(p.count / 24);
      p.avgWeekdayEnergy = p.totalkWhGroup / Math.floor(p.count / 24);
    }
    else {
      p.avgWeekday = p.price / p.count;
      p.avgWeekdayEnergy = p.totalkWhGroup / p.count;
    }
  }
  else {
    p.avg = 0;
    p.avgWeekday = 0;
  }
  return p;
}
```

- `canvasSize(targetElement)`: Encargada de calcular el tamaño de un DIV o SVG en pixels. El dato que devuelve es un vector de 2 elementos que indican el ancho y el alto del elemento pasado como argumento. Esta función es una parte vital, sobre la que se sustenta toda la aplicación, ya que ha sido diseñada para que sea responsiva y se adapte a cualquier tamaño

de pantalla siempre que esta guarde una relación de aspecto 4:3, que es la que tenía el equipo sobre la que fue diseñada.

- `hover(hoverD)`, `mouseOut()`: Se funciones complementarias y se activan cuando el puntero del ratón se pone encima, o se quita, de los elementos del "histograma 2D", mostrando información relativa a ese dato.
- `Crear_Scatter(dimension, targetDiv)`: Crea un scatter para representar la correspondencia entre tres atributos distintos. Recibe como argumentos una dimensión cualquiera del conjunto de datos (es indiferente para para función) y el *id* del SVG del destino. Nota: No ha sido usada en la versión final del proyecto.

Tras comentar someramente la utilidad y uso de cada función, se hará lo propio con los comandos que han sido más útiles y repetidos a la hora de crearlas. El primero de ellos no puede ser otro que el mecanismo de selección de objetos DOM que nos brinda D3, y este es el comando *select*. Un ejemplo simple de selección de todos los rectángulos de una determinada clase dibujados en un determinado SVG sería:

```
var rectangulosSeleccionados = d3.select('#ID_del_SVG').selectAll('rect.Nombre_de_Clase');
```

Si ahora quisiéramos cambiar el estilo o apariencia de esos rectángulos, por ejemplo "pintarlos" de rojo:

```
rectangulosSeleccionados.style('fill', 'red');
```

Todo esto no es nada novedoso ya que la librería *jQuery*<sup>16</sup> ya permitía hacerlo mediante mecanismos de selección parecidos. Donde radica la fuerza de D3, es a la hora de crear o dibujar automáticamente cientos o miles de elementos con muy poco esfuerzo. Supongamos que tenemos un fichero de datos en .csv o .json, que contiene una serie de elementos con sus correspondientes atributos, y lo leemos:

```
d3.json("Datos.json", function (error, data) {  
var datosLeidos = data;
```

Lo primero que se debe hacer para poder dibujar o representar ese número objetos, es crear un "lienzo" (SVG) sobre el que dibujarlos y darle un nombre identificativo o ID. En este ejemplo lo crearemos cuadrado de 500 pixels de lado:

```
var lienzo = d3.select('body').append('svg').attr('id', 'NombreLienzo')  
    .attr('height', '500').attr('width', '500');
```

---

<sup>16</sup>Biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

Como se puede ver, el comando `.attr()` sirve para asignar atributos a un determinado objeto del DOM, en este caso fueron el "id", la altura "height" y la anchura "width".

Ahora que se ha creado el SVG, imaginemos que el conjunto de datos en formato .json, que leímos con anterioridad, tiene el aspecto de la tabla 3.1 que se muestra a continuación.

ID trabajador	edad	salario (€/Año)	años trabajados
33837	37	29.000	8
23400	30	21.000	4
43837	50	40.000	27
35312	42	34000	15
39912	46	36.000	20
36488	26	29.000	6
44231	48	41.000	12
53621	33	34000	9
33837	21	29.000	1
10361	60	62.000	40
78001	66	50.000	31
66721	44	34000	10

Tabla 3.1: Estructura de los datos del ejemplo en D3

Conociendo esto, podemos representar los círculos a lo largo y alto del SVG en función de, por ejemplo, la edad del trabajador (eje horizontal X) y los años trabajados (eje vertical Y). Además, haremos que su radio sea proporcional al salario. Para ello, necesitamos crear escalas que "escalen" los valores de la tabla a al tamaño en pixels del SVG. Esto se hace de la siguiente manera:

```
var Xextent = d3.extent(datosLeidos, function (d) { return d.edad });
var Xescala = d3.scale.linear().domain(Xextent).range([20, 480]);

var Yextent = d3.extent(datosLeidos, function (d) { return d.aniosTrabajados });
var Yescala = d3.scale.linear().domain(Yextent).range([480, 20]);

var Rextent = d3.extent(datosLeidos, function (d) { return d.salario });
var Rescala = d3.scale.linear().domain(Rextent).range([5, 20]);
```

Como se puede observar, para cada una de las 3 escalas, primero se calculan los valores mínimo y máximo con el método `extent()` y, posteriormente, esos valores se le pasan a `domain()` para indicar el dominio de la escala. Por último, tras indicar el dominio, indicamos el rango en pixels al cual se tiene que adaptar, en este caso de 20 a 480 (dejamos 20 pixels de margen por cada lado), usando el comando `range([min,max])`.

Se crean los ejes horizontal y vertical (edad del trabajador y años en la empresa respectiva-

mente) y se les llama para que se dibujen:

```
var XAxis = d3.svg.axis().scale(Xescala).orient("bottom").ticks(6);
    var YAxis = d3.svg.axis().scale(Yescala).orient("left").ticks(6);

    d3.select('#NombreLienzo').append("g")
      .attr("class", "Xaxis")
      .attr("transform", "translate(0,460)")
      .call(XAxis);

    d3.select('#NombreLienzo').append("g")
      .attr("class", "Yaxis")
      .attr("transform", "translate(40,0)")
      .call(YAxis);
```

Una vez creados el lienzo (SVG) y las escalas, podemos dibujar, fácilmente, tantos círculos como elementos tenga *"datosLeidos"*, en este caso sencillo son sólo 4.

```
var circulos = d3.select('#NombreLienzo').selectAll('circle').data(datosLeidos)
  .enter().append('circle')
  .attr("cx", function (d) { return Xescala(d.edad); })
  .attr("cy", function (d) { return Yescala(d.aniosTrabajados); })
  .attr("r", function (d) { return Rescala(d.salario); })
  .style('fill', 'red');
```

Esto lo que hace es seleccionar el SVG indicado, dentro de este, todos los círculos (no debería haber ninguno todavía) y después, crear tantos como elementos tiene la matriz que se pasa como argumento al método *data()*. Se puede observar también como se definen los atributos de cada círculo con *attr()*, tal y como ya se había comentado anteriormente. Estos atributos son *cx* y *cy* para las coordenadas del centro y *r* para el radio.

El código completo del ejemplo sería el siguiente:

```
<!DOCTYPE html>
<script src="https://cdnjs.cloudflare.com/ajax/libs/d3/3.5.6/d3.min.js" charset="utf-8"
  ></script>
<head>
  <title></title>
</head>
<body>
  <script>

d3.json("datos.json", function (error, data) {
var datosLeidos = data;

var lienzo = d3.select('body').append('svg').attr('id', 'NombreLienzo')
```

```

        .attr('height', '500').attr('width', '500').style('background', '
        white') ;

var Xextent = d3.extent(datosLeidos, function (d) { return d.edad });
var Xescala = d3.scale.linear().domain(Xextent).range([40, 460]);

var Yextent = d3.extent(datosLeidos, function (d) { return d.aniosTrabajados
});
var Yescala = d3.scale.linear().domain(Yextent).range([460, 40]);

var Rextent = d3.extent(datosLeidos, function (d) { return d.salario });
var Rescala = d3.scale.linear().domain(Rextent).range([5, 20]);

var XAxis = d3.svg.axis().scale(Xescala).orient("bottom").ticks(6);
var YAxis = d3.svg.axis().scale(Yescala).orient("left").ticks(6);

d3.select('#NombreLienzo').append("g")
  .attr("class", "Xaxis")
  .attr("transform", "translate(0,460)")
  .call(XAxis);

d3.select('#NombreLienzo').append("g")
  .attr("class", "Yaxis")
  .attr("transform", "translate(40,0)")
  .call(YAxis);

var circulos = d3.select('#NombreLienzo').selectAll('circle').data(datosLeidos)
  .enter().append('circle')
  .attr("cx", function (d) { return Xescala(d.edad); })
  .attr("cy", function (d) { return Yescala(d.aniosTrabajados); })
  .attr("r", function (d) { return Rescala(d.salario); })
  .style('fill', 'red');
  });
</script>
</body>
</html>

```

El resultado del código anterior sería el que muestra la figura 3.14. No es más que un scatter en el que se podría apreciar que la relación entre la edad (eje X) y los años trabajados en la empresa (eje Y) sigue una relación prácticamente lineal y cuyos salarios (el radio de los círculos) se incrementa a media que lo hacen las otras 2 variables.

En la tabla 3.2 se muestra un resumen con todas las funciones creadas para el proyecto.

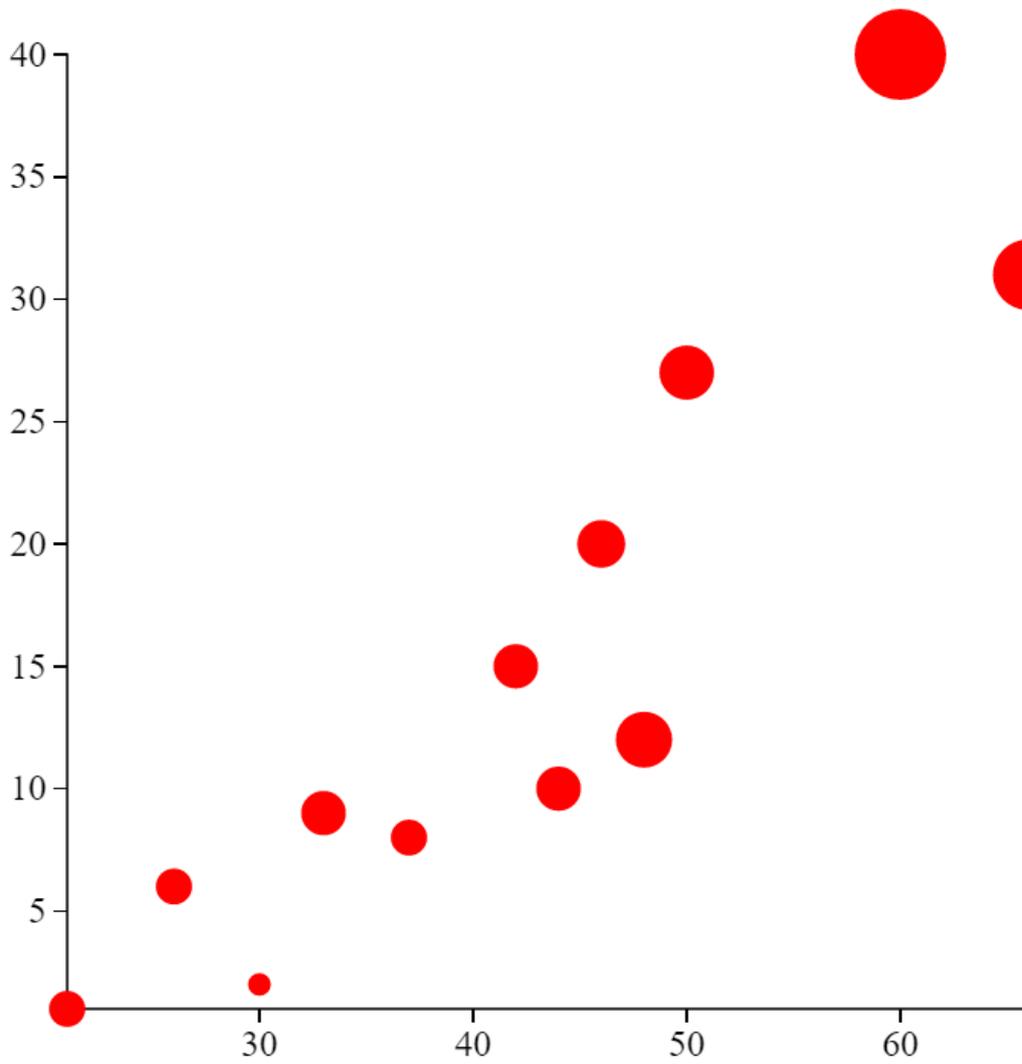


FIGURA 3.14. Ejemplo de representación sencilla en D3 a partir del código de ejemplo mostrado.

Función	Descripción
Crear_Dimensiones(DataArray)	Usada para crear las dimensiones
Crear_Histogramas()	Crea todos los histogramas con los parámetros iniciales
EliminarHistos()	Elimina todos los histogramas
Crear_Histo2D(dimension, grupo, targetDiv)	Crea un histograma en forma de scatter cuya dimensión es la combinación de 2 dimensiones de Crossfilter
Crear_Histo(dimension, grupo, targetDiv, ordinal, abajo, gruposConReduce, usarReduce)	Crea un histograma con eje cuantizado u ordinal, usando o no el método reduce de Crossfilter
update()	Usada para actualizar los histogramas tras cualquier interacción por parte del usuario
Crear_Spreadsheet(dimension, target_Div, max)	Crea una tabla cuyas columnas son los atributos de la matriz de datos original
Crear_controles()	Crea todos los botones de la aplicación
Crear_valoresMax(target_Div)	Muestra los valores máximos de ciertos histogramas
modifytariff()	Crea los sliders o la tabla dependiendo del estado actual
crear_slider(target_Div, grupo)	Crea un grupo de sliders que ocupe todo el ancho de la página
slideFunc(evt, value)	Se ejecuta al modificar el valor de los sliders
resetPrice()	Reseta el valor de cada slider
ordenarActiva(), ordenarCosPhi(), ordenarHora()	Ordenan la tabla en función de ese atributo
CrearGrupos(matrizGeneral)	Crea dos grupos a partir de la matriz de datos original
agruparCombinaciones(grupo1, grupo2, matrizGeneral)	Agrupar, los dos grupos que crea la función <i>Crear-Grupos</i> , en todas las combinaciones posibles
agregadosPrice(grupos)	Cambia el título de los histogramas a datos agregados de coste en €
agregadosEnergy(grupos)	Cambia el título de los histogramas a datos agregados de energía en kWh o MWh
reduceAdd(p, v), reduceRemove(p, v), reduceInit(p, v)	Funciones reduce de Crossfilter
canvasSize(targetElement)	calcula el tamaño del Div indicado
hover(hoverD), mouseOut()	Muestran o quitan información cuando se pasa por encima de los elementos del "histograma2D"
Crear_Scatter(dimension, targetDiv)	Crea un scatter con 3 atributos (No usado en el proyecto)

Tabla 3.2: Tabla de funciones del proyecto.

## RESULTADOS

**E**n este capítulo se comenzará explicando detalladamente cada uno de los casos estudiados sobre el comportamiento, a nivel de consumo energético, de las universidades de Química y Economía pertenecientes a la ciudad de Oviedo. Posteriormente, se analizará todo el *feedback* recibido tras un *user study* ( estudio de usuario ) realizado a ocho personas cercanas.

### 4.1 Casos de estudio

Los casos que se han estudiado son un total de *seis* con sus *tres* variantes, seleccionado los dos edificios, sólo el de química o sólo el de economía. El procedimiento es similar en todos ellos; se selecciona la región objeto de estudio y se observa el comportamiento en el resto de las gráficas. Los seis casos de estudio se listan a continuación:

1. Identificación de los consumos más frecuentes
2. Identificación de cuándo se producen los mayores consumos
3. Identificación de cuándo se producen los menores consumos
4. Identificación de cuándo se producen los peores factores de potencia y las consecuencias de esto
5. Identificación de consumos residuales
6. Identificación del cambio en la factura al hacer modificaciones en la tarifa horaria



FIGURA 4.1. Captura que muestra los consumos más frecuentes en ambas universidades ( el resto de las capturas se encuentra al final del capítulo ).

#### 4.1.1 Identificación de los consumos más frecuentes

Como se puede ver en la figura 4.1, hay dos regiones claramente predominantes. Una está entre los 45kWh y la otra se encuentra en torno a los 145kWh. Esto puede ser una indicación de que cada una de esas regiones nos está indicando los consumos más frecuentes de cada facultad. Para comprobarlo, no tenemos más que seleccionar, en el histograma "Buildings", una de las dos y comprobar que una de las regiones desaparece.

Las figuras 4.2(a) y 4.2(b) confirman la sospecha, mostrándonos que la región entorno a los 45kWh pertenece a la facultad de Economía, mientras que la de 145kWh la perteneciente a la de Química.

#### 4.1.2 Identificación de cuando se producen los mayores consumos

En este caso, lo que se hizo fue seleccionar, con el *brush*<sup>1</sup> del propio histograma de energía consumida, los consumos más altos, concretamente los 125 mayores. Esto, como bien muestra la

<sup>1</sup>Es una función de selección que nos permite escoger una porción de la gráfica sin más que hacer clic en una determinada zona de esta y arrastrar en la dirección que se desee.

figura 4.3(a) , nos indica varias cosas de gran interés que se listan a continuación:

- *Todos los consumos* pertenecen a la facultad de Química.
- Se producen durante la mañana, concretamente de 10:00 a 14:00, siendo la hora habitual de mayor consumo las 12.
- Se dan en los días lectivos de lunes a jueves, apareciendo el viernes casi una tercera parte de las veces que en estos.
- Hay un mes que destaca por encima de todos los demás, febrero, que, a pesar de tener menos días que el resto, tiene 70 de las 125 mediciones seleccionadas. Por el contrario junio, julio, agosto y septiembre no parecen haber registrado ninguna.
- Los factores de potencia muestran unos buenos valores al no bajar de 0.97.
- Se dan (los mayores consumos), íntegramente, en días no festivos; algo obvio por otra parte.

Al encontrarnos con que todos los consumos pertenecen a la facultad de Química, buscamos cuales son los mayores de Economía. El resultado se puede ver en la figura 4.3(b). Lo primero que salta a la vista es que el mayor consumo registrado es 322.66kWh, lo que son 113kWh menos que el mayor registrado en Química<sup>2</sup>. En lo referente a los puntos comentados en el caso anterior (el de Química), podemos decir:

- Las horas del día, al igual que los factores de potencia, se muestran similares
- El histograma de los días de la semana muestra un curioso decremento a medida que nos vamos acercando al fin de semana, en el cual no hay ninguna medición registrada
- En el caso de los meses del año, la mayor parte de los mayores consumos se concentran en los *tres* últimos meses (octubre, noviembre y diciembre) quedando a cero, como ocurría en la facultad de Química, los meses de junio, julio, agosto y septiembre

### 4.1.3 Identificación de cuándo se producen los menores consumos

Pero la identificación de estos consumos seleccionamos los más bajos en el histograma de energía consumida. La figura 4.4(a) nos muestra, al contrario de lo que pasaba con los mayores consumos, los cuales pertenecen, exclusivamente, a la facultad de Economía. Además, podemos extraer la siguiente información:

<sup>2</sup>Los valores máximos y mínimos se indican en la tabla bajo los histogramas

- Se producen a todas las horas del día, siendo más frecuentes entre las 9:00 y las 19:00 horas.
- Ocurren la mayoría de las veces en el fin de semana.
- El mes que acapara la inmensa mayoría de los consumos más bajos es agosto, mes en el que Universidad de Oviedo tiene actividad mínima.
- El factor de potencia en estos casos es sumamente pobre, llegando a bajar hasta los 0.47.

Para identificar cuales son en la facultad de Química, la seleccionamos en la gráfica *Building*, haciendo click en la barra correspondiente a la letra *C* de *Chemistry*. El resultado se puede ver en la figura 4.4(b), arrojando similares resultados, salvo dos excepciones. Las horas a las que se producen están distribuidas de una forma más homogénea alcanzando, si cabe, las mayores cotas en horario de madrugada. La otra diferencia destacable se produce en los factores de potencia que, como se comentará en la siguiente sección, aunque son malos, nunca bajan de 0.75.

#### 4.1.4 Identificación de cuándo se producen los peores factores de potencia y consecuencias

En la figura 4.5(a) se puede observar como los valores con la mayor penalización tarifaria (menores o iguales a 0.75) sólo se dan en la facultad de Economía<sup>3</sup>, llegando incluso a bajar de 0.5, lo que es un factor de potencia sumamente pobre. Por el contrario, la facultad de química mantiene unos niveles mucho mejores, al no bajar prácticamente nunca de 0.85. Se puede ver en la figura 4.5(b) que apenas *trescientas* horas, en el transcurso de todo un año<sup>4</sup>, están por debajo de ese valor. Concretamente, un 3.5% de las horas.

Esta desigualdad en el comportamiento del factor de potencia entre ambas escuelas, y la consiguiente penalización que valores bajos de este acarrea, derivan en unos costes más altos del kWh para la facultad de economía, pasando de los  $0.1289\text{€}/kWh$  de química ( figura 4.6(b) ) a los  $0.1381\text{€}/kWh$  ( figura 4.6(a) ). Si aplicamos esta diferencia de precios al consumo total de energía de la escuela de economía de todo el año ( figura 4.7 ), tendremos el gasto que provoca su pobre factor de potencia.

En la ecuación 4.1 se puede ver un sobrecoste, en el plazo de un año, de unos *siete mil euros* con respecto al que tendría si mantuviera los mismos niveles que la facultad de químicas.

$$(4.1) \quad (0.1381kWh - 0.1289kWh) * 751.85Mwh = 6,917.02\text{€}$$

<sup>3</sup>En la facultad de Química se han registrado ocho horas, a lo largo de todo un año, cuyo valor baja hasta 0.75; cantidad despreciable frente al total.

<sup>4</sup>Un año tiene 8760 horas.

En referencia a cuándo se dan esos valores bajos del factor de potencia en cada facultad, se obtienen por un lado ciertas similitudes:

- Siempre se dan en los consumos más bajos.
- Se distribuyen a lo largo de todas las horas del día con especial predominancia en la madrugada (hasta las 6:00 am).

Igualmente también se observan algunas disparidades:

- En cuanto a los días, se aprecia que en la escuela de Economía se dan mayoritariamente los fines de semana, mientras que en la de Química se dan indistintamente sin, aparentemente, ningún patrón definido.
- De lo anterior deriva que en Economía se den prácticamente por igual tanto en días festivos (incluyendo domingos como es evidente) como en el resto de días, mayoritariamente sábados; en el caso de Química, los registros en días festivos son prácticamente despreciables frente al resto.
- La última discrepancia la encontramos al examinar el histograma de los meses, el cual nos arroja un curioso hecho. Mientras que en la Escuela de Química se dan casi íntegramente en agosto y, en mucha menor medida, septiembre (suponemos que los primeros días de este), en la Escuela de Economía se da en los primeros 5 meses del año.

#### 4.1.5 Identificación de consumos residuales

Para averiguar estos consumos sólo hay que seleccionar, en el histograma *Holiday*, la barra correspondiente al número 1 o *fiesta* == *True* expresión típica del ámbito de la programación. Una vez hecho esto, obtenemos la captura representada en la figura 4.8(a), la cual nos muestra una distribución claramente bimodal en el histograma de potencia activa. Como es obvio, esto se atribuye a los consumos residuales de ambas universidades. Para confirmarlo, se selecciona una de ellas en el histograma *Building*, con el fin de que desaparezca uno de los dos máximos locales o modos. La figura 4.8(b), que muestra los de la facultad de química, confirma lo supuesto y nos aclara los siguiente:

- El consumo de ésta, entorno a 140kW, es *tres veces mayor* que el de la de Economía, que se sitúa alrededor de los 45kW.

#### 4.1.6 Identificación del cambio en la factura al hacer modificaciones en la tarifa horaria

Debido a que en este apartado, las posibilidades de configuración de distintas tarifas horarias se pueden considerar infinitas, se ha optado por elegir las cuatro horas del tramo horario más caro y reducir su precio *1 céntimo de euro*. Concretamente se ha pasado de los  $0.104\text{€}/kWh$  a los  $0.92\text{€}/kWh$ . Es un cambio que, a pesar de ser pequeño, es excelente para mostrar la incidencia que tiene sobre los costes a final de año.

Las figuras 4.9(a) y 4.9(b) muestran lo comentado. La primera nos indica los costes totales con la actual tarifa eléctrica ( $325,281.70\text{€}$ ) mientras que la segunda muestra los obtenidos tras la modificación de esta ( $318,381.83\text{€}$ ). Esto nos arroja un ahorro cercano a los  $7,000\text{€}$  anuales que, prorrateado mensualmente, son  $530\text{€}$ .

## 4.2 Estudio de usuario

Para extraer conclusiones sobre la aplicación creada, siempre es conveniente recibir *feedback* de personas ajenas a esta con el fin de mejorarla, si no en la primera versión, sí de cara al futuro. Por ello, se llevó a cabo un user study (estudio de usuario) que ha requerido la colaboración de familiares y amigos cercanos. El total de personas ha sido de *ocho*, siendo su distribución por edades y sexos la siguiente:

- Cinco personas en la franja 50-65 años, repartidas en 3 mujeres y dos hombres, todas ellas totalmente ajenas al ámbito tecnológico o eléctrico.
- Tres en la franja 30-40 años de las cuales hay una mujer. Todas ellas habituados al uso de la tecnología, pero sólo uno de los hombres con conocimientos eléctricos.

A todas ellas se les instó a usar la aplicación creada sin decirles absolutamente nada, simplemente que leyeran las instrucciones de uso que esta misma muestra en el propio navegador web. Las personas que no sabían inglés, o tenían problemas para entenderlas en ese idioma, recibieron una traducción oral de estas. Además, se les pidió que dijeran todo lo que no entendieran o no les gustase, así como que aportaran posibles mejoras con el fin de, tras estudiar su idoneidad, poder implementarlas en futuras versiones.

La meta principal de este estudio era extraer conclusiones, fundamentalmente, sobre *tres* aspectos clave:

- *Facilidad de uso*; hace referencia a la facilidad con que un usuario puede utilizar una herramienta fabricada por otras personas con el fin de alcanzar un cierto objetivo.

- *Utilidad instantánea*; que los resultados mostrados sean obvios al usuario e inmediatos.
- *Idoneidad del diseño*; un diseñador ha de conocer las tendencias actuales, la teoría de la composición, el estudio de la armonía de colores, las diferentes emociones que produce la tipografía, etc.

Para este propósito, se llevo a cabo una sencilla encuesta en la que se pedía a los usuarios que puntuaran de 0, muy malo, a 5, muy bueno, cada uno de estos 3 apartados.

Los resultados obtenidos, los cuales se plasman en la tabla 4.1, nos arrojan lo siguiente:

- El apartado *Facilidad de uso* obtiene unas valoraciones muy dispares, moviéndose entre el 3 y el 5. Su nota media total es de 3.75, estando más cerca de "buena" que de "normal", mientras que si acotamos a la franja de edad más joven ( por debajo de los 40 años ), la nota es de 4.33 ("buena"). En la franja de mayor edad ( 50 - 65 años ) se obtiene un 3.4.
- El apartado *Utilidad instantánea* es el apartado que mejores calificaciones obtiene, alcanzado una nota media de 4.875, muy cerca del 5 ("muy buena") debido a la elección, casi por unanimidad, de este valor. En este aspecto, se aprecia que la edad y sexo es indiferente a la hora de sacar conclusiones, de forma rápida, sobre las selecciones efectuadas por cada uno de los usuarios.
- El apartado *Idoneidad de diseño*, al igual que el de "*Facilidad de uso*", muestra unos resultados heterogéneos, aunque siempre entre "normal" y "muy bueno", alcanzando una nota media de 3.875. Se aprecia, además, un patrón de notas más bajas dentro del sector femenino, no siendo así en lo que respecta a la franja de edad.

Facilidad de uso	Utilidad instantánea	Idoneidad de diseño	Edad	Sexo
4	5	4	64	M
3	5	3	63	F
4	5	5	60	F
3	4	3	59	F
3	5	4	55	M
3	5	3	31	F
5	5	4	33	M
5	5	5	36	M

Tabla 4.1: Resultados del *User Study*. En ella se valora de 0 a 5, siendo 0 malo y 5 muy bueno, cada uno de los tres apartados.

### 4.2.1 Conclusiones extraídas y sugerencias aportadas por los usuarios

Tras la prueba de la aplicación, así como también durante el uso de ésta, se recibieron una serie de preguntas y sugerencias, casi en su totalidad, centradas en el aspecto de la "*Facilidad de uso*", que se pasan a listar a continuación:

- Sólo dos de las personas fueron las que no mostraron la más mínima duda sobre como proceder, entendiendo el comportamiento de la función de selección de cada histograma, así como el de todos los botones que preceden la tabla, tras un muy corto tiempo de prueba.
- Otras personas (dos), a pesar de entender perfectamente el método de selección, no entendieron, sin una corta explicación por mi parte, para que servían todos los botones.
- El resto, cuatro personas, al no estar habituadas al uso de la tecnología (según sus propias palabras), argumentaron que no sabían muy bien donde tenían que hacer click o para qué servían los botones. Tras una muestra de ejemplo por mi parte, menos de *un minuto*, mostraron que lo habían entendido a la perfección.
- Se sugirió que, para facilitar el entendimiento de la función de cada botón, se mostrara ésta, a modo de ventana flotante, al pasar con el puntero del ratón por encima de ellos.

A pesar de los datos obtenidos en el estudio de usuario y la posterior encuesta, se es plenamente consciente de que la muestra es muy pequeña como para que sea representativa. Un factor más a añadir a esto, es la posible "*deformación*" de las puntuaciones obtenidas, ya sea consciente como inconscientemente, debido a la relación de parentesco o amistad.

## 4.2. ESTUDIO DE USUARIO

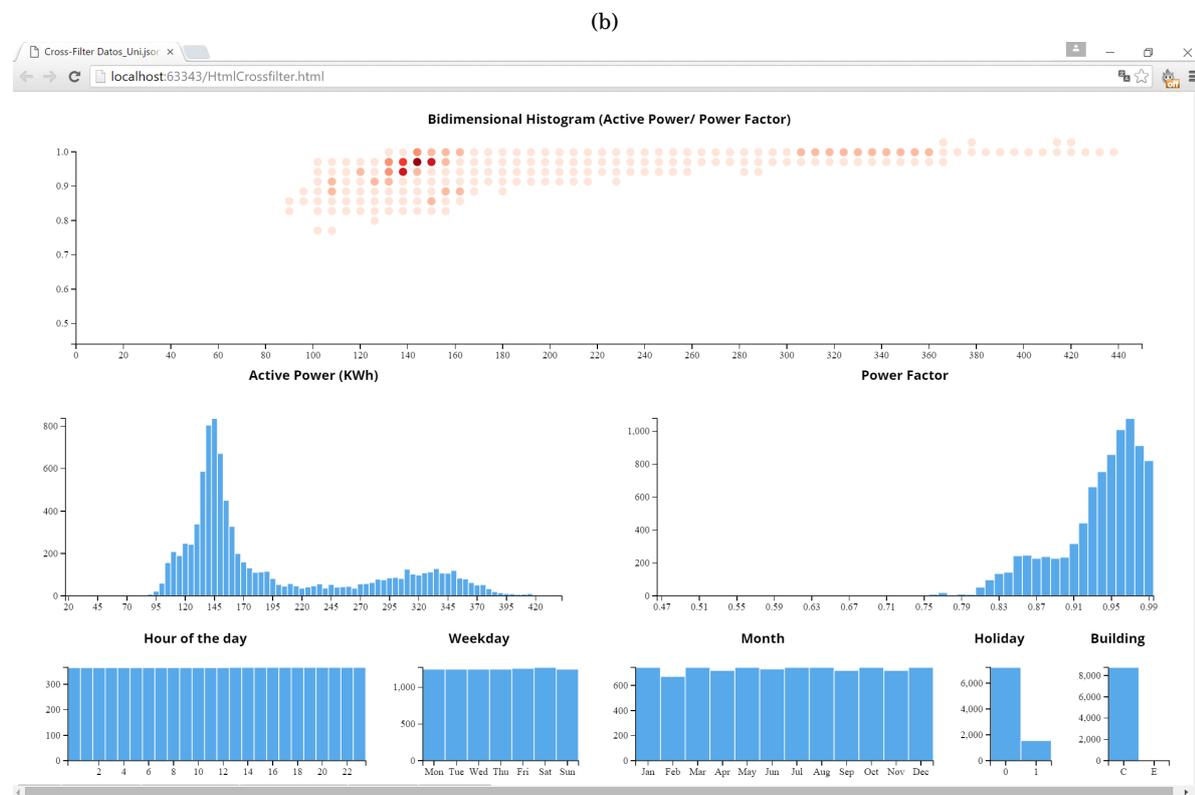
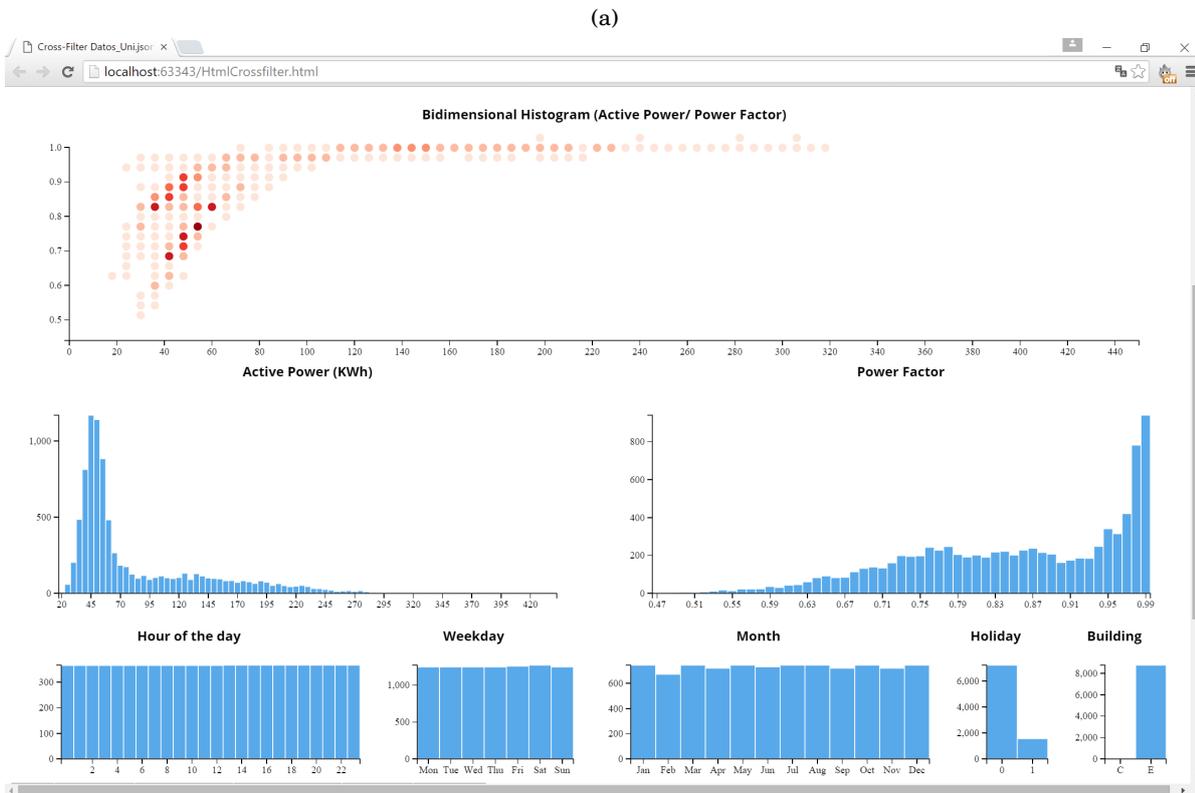
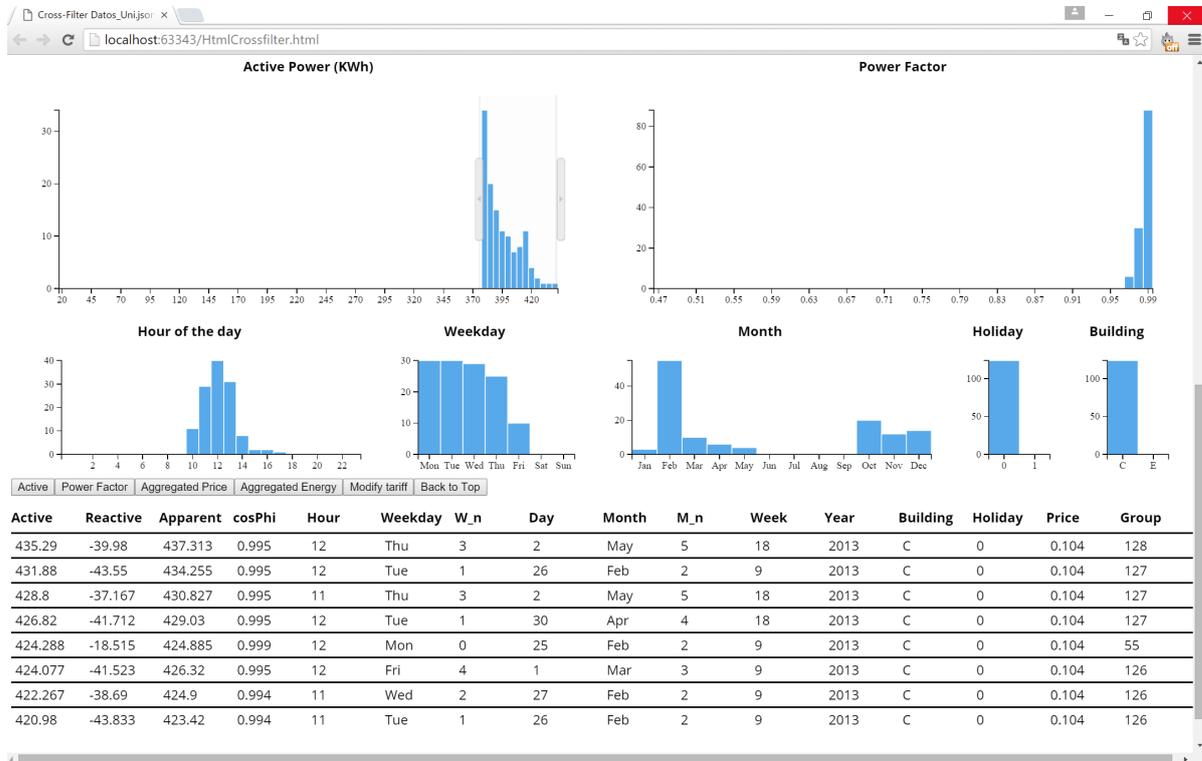


FIGURA 4.2. (a) Captura que muestra los consumos más frecuentes de economía; (b) Captura que muestra los consumos más frecuentes de química.

## CAPÍTULO 4. RESULTADOS

(a)



(b)

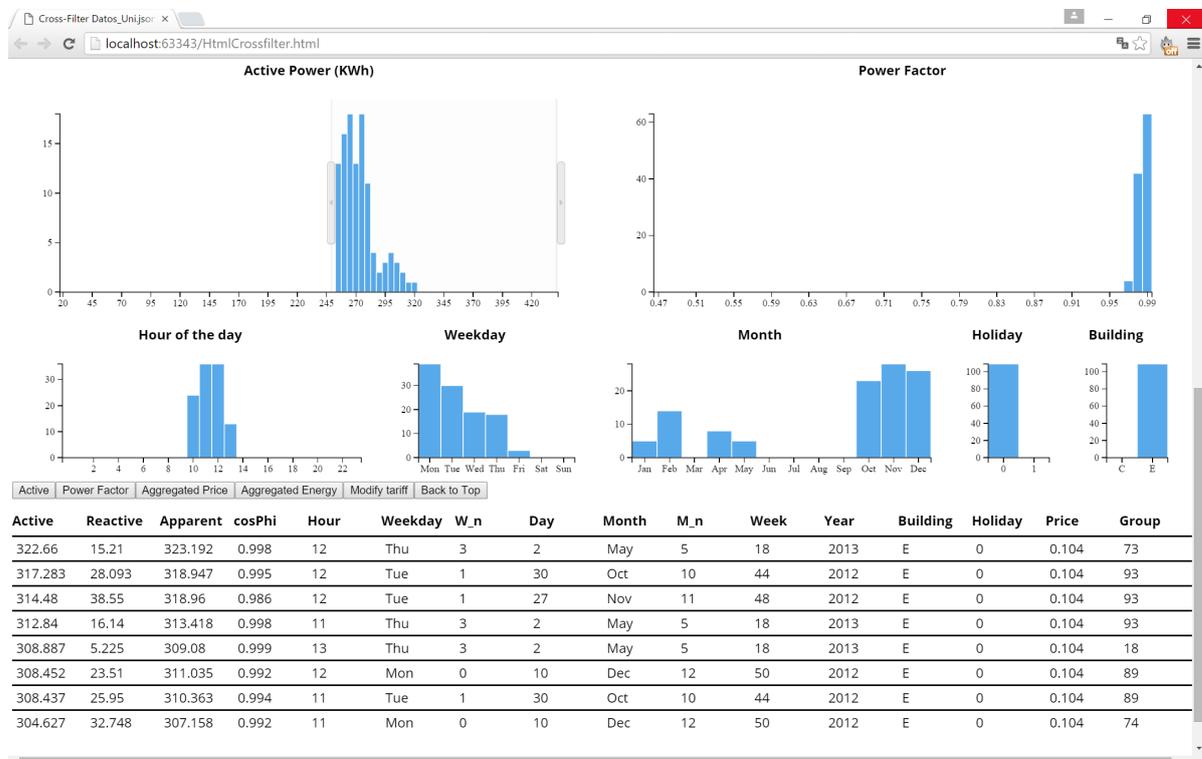
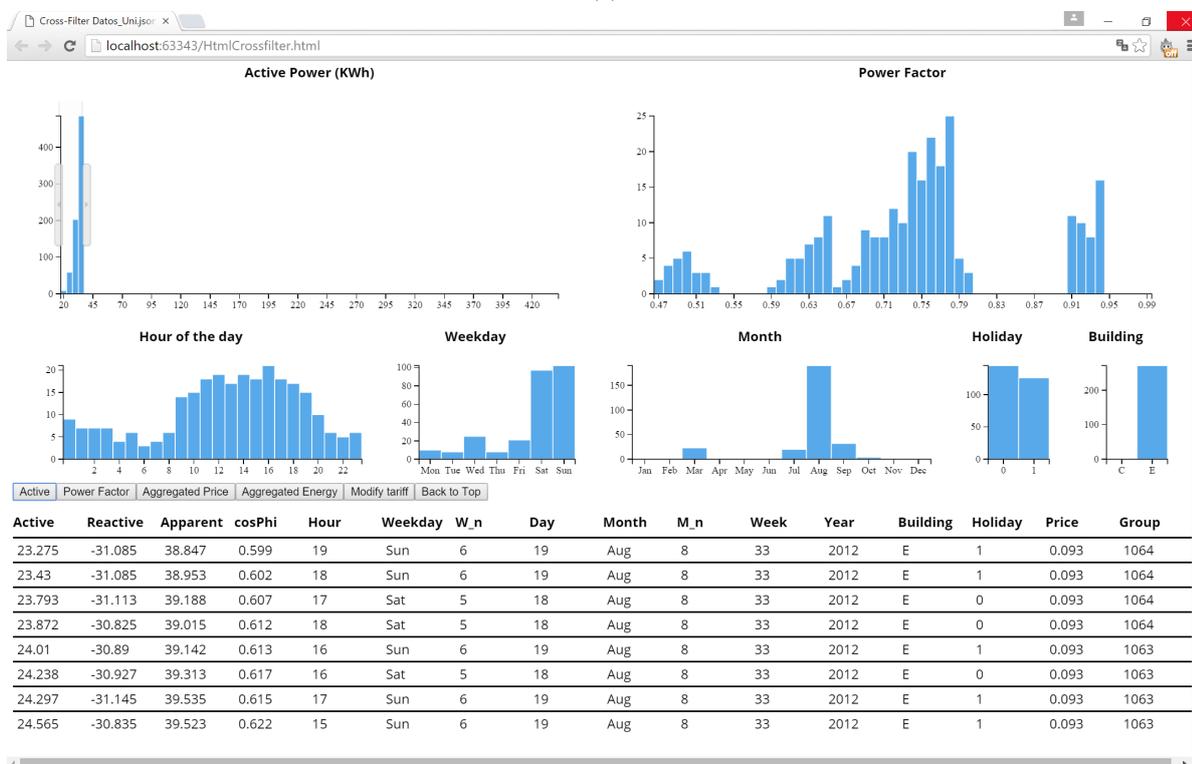


FIGURA 4.3. (a) Captura que muestra los mayores consumos en la facultad de química; (b) mayores consumos en economía.

## 4.2. ESTUDIO DE USUARIO

(a)



(b)

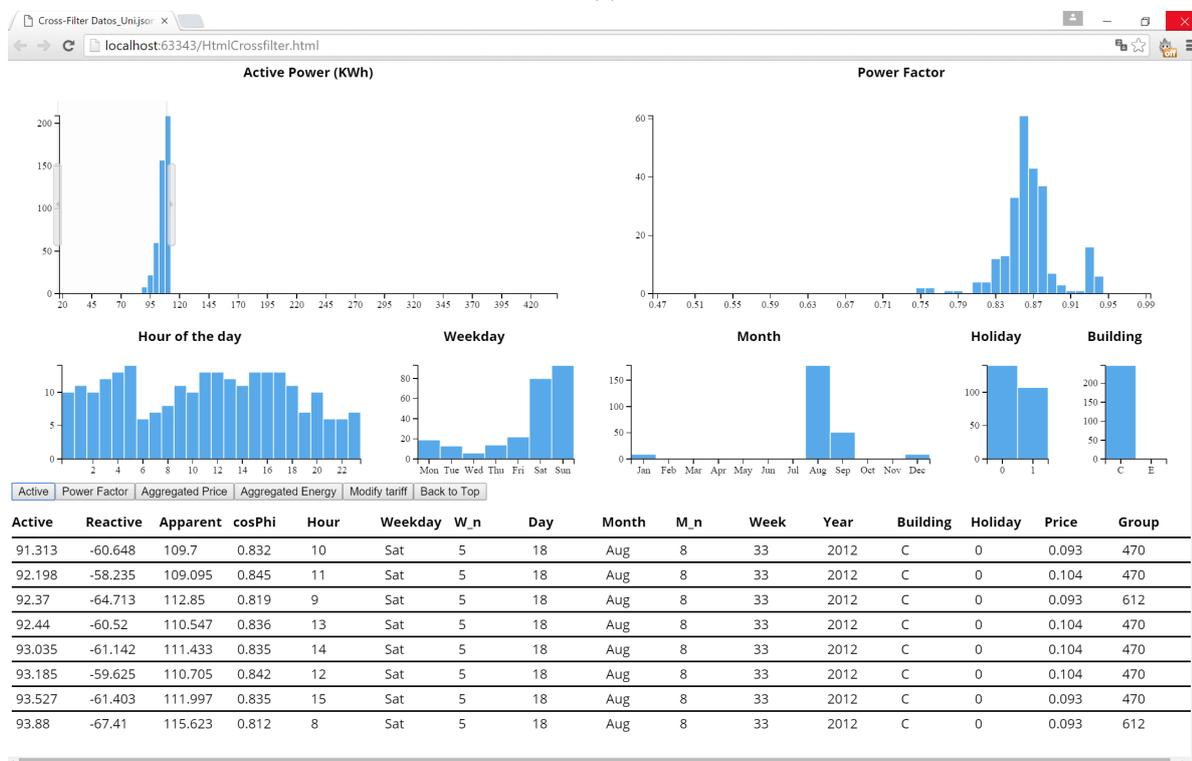
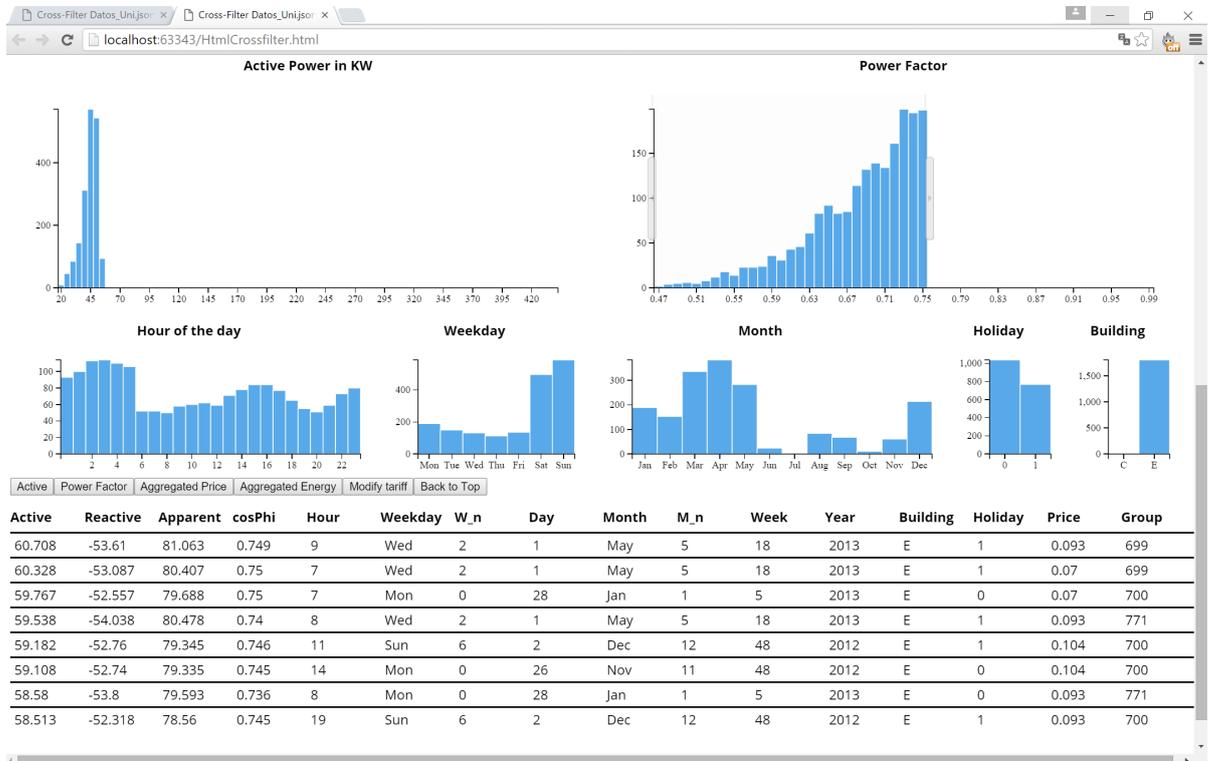


FIGURA 4.4. (a) Captura que muestra los menores consumos de economía; (b) menores consumos de química.

## CAPÍTULO 4. RESULTADOS

(a)



(b)

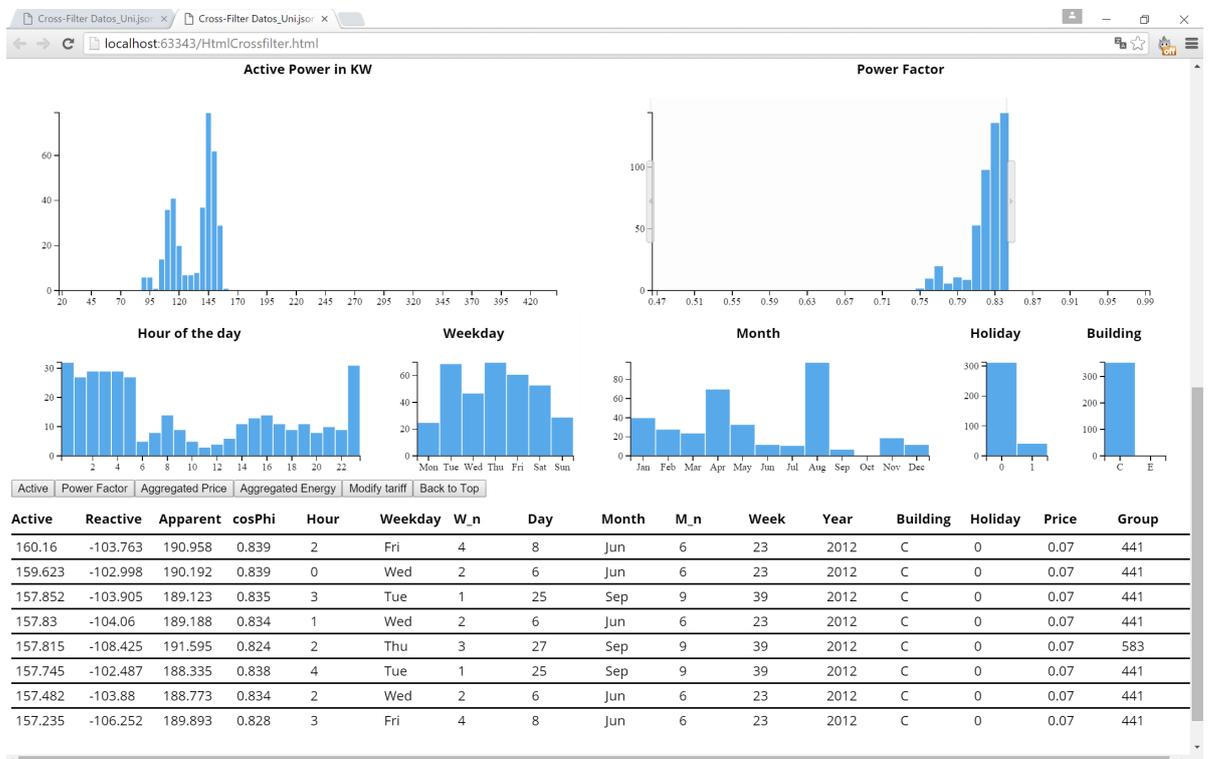
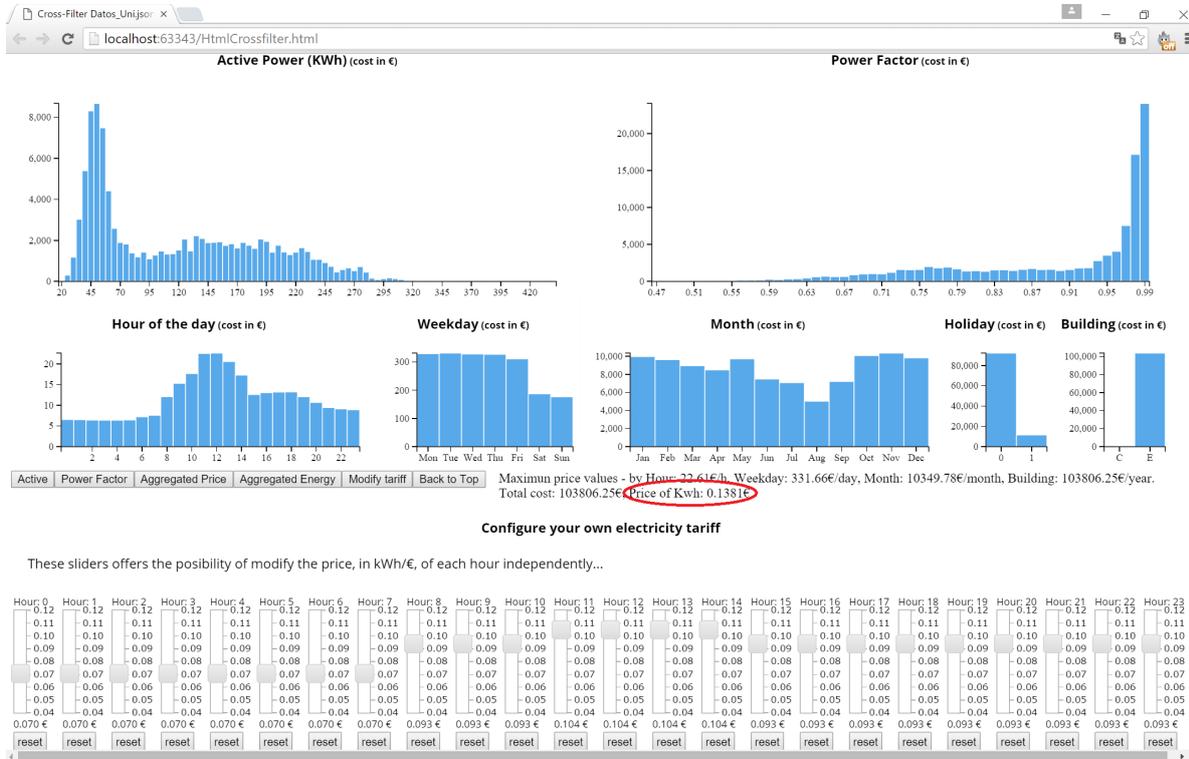


FIGURA 4.5. (a) Captura de los factores de potencia iguales o menores de 0.75; (b) Captura de los peores factores de potencia en la facultad de química.

## 4.2. ESTUDIO DE USUARIO

(a)



(b)

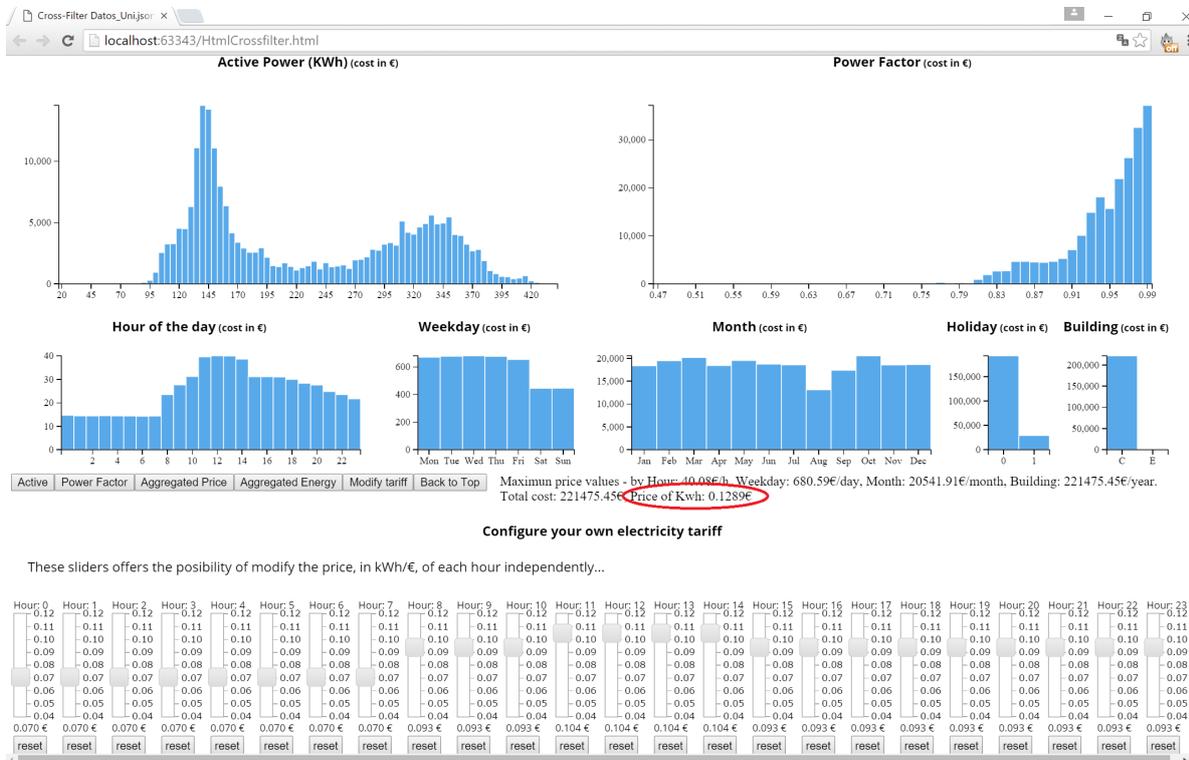


FIGURA 4.6. (a) Captura que muestra el precio kWh de la facultad de económicas; (b) Captura que muestra el precio kWh de la facultad de químicas.

## CAPÍTULO 4. RESULTADOS

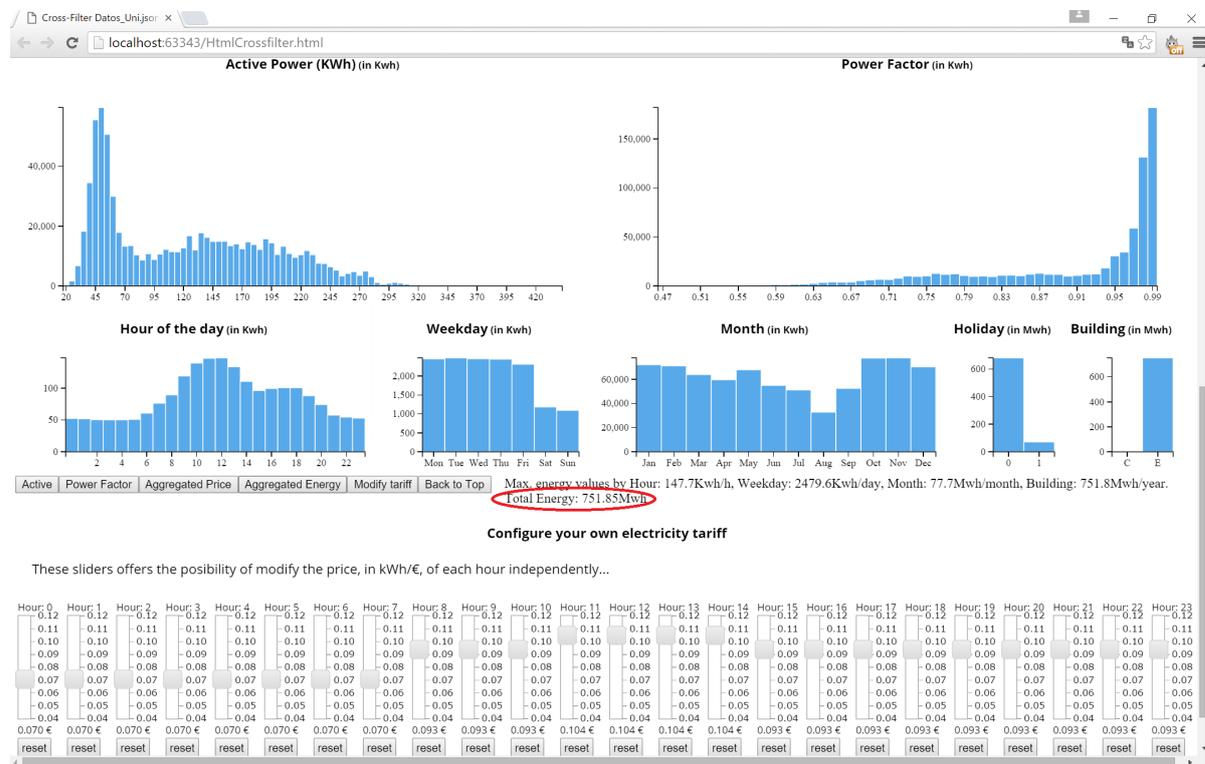


FIGURA 4.7. Captura que muestra el consumo total de la facultad de económicas.

## 4.2. ESTUDIO DE USUARIO

(a)



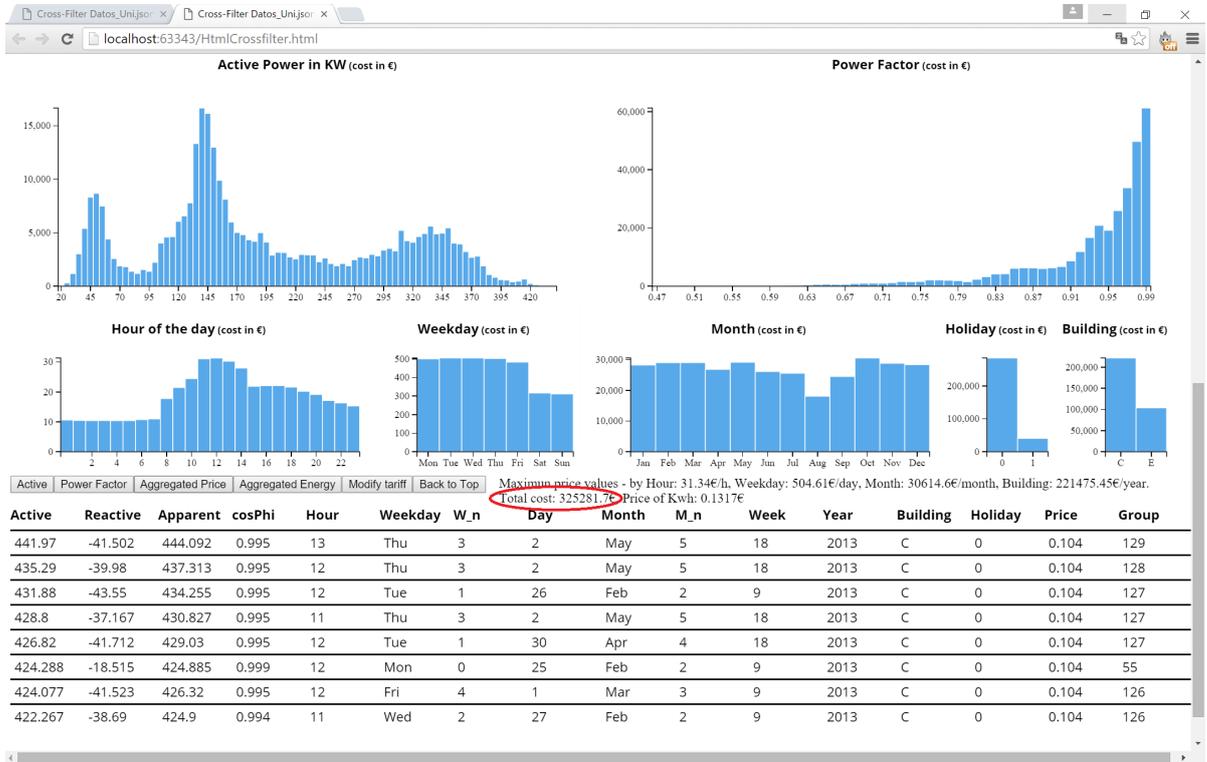
(b)



FIGURA 4.8. (a) Captura del consumo residual de la Escuela de Economía; (b) Captura del consumo residual de la Escuela de Química.

# CAPÍTULO 4. RESULTADOS

(a)



(b)

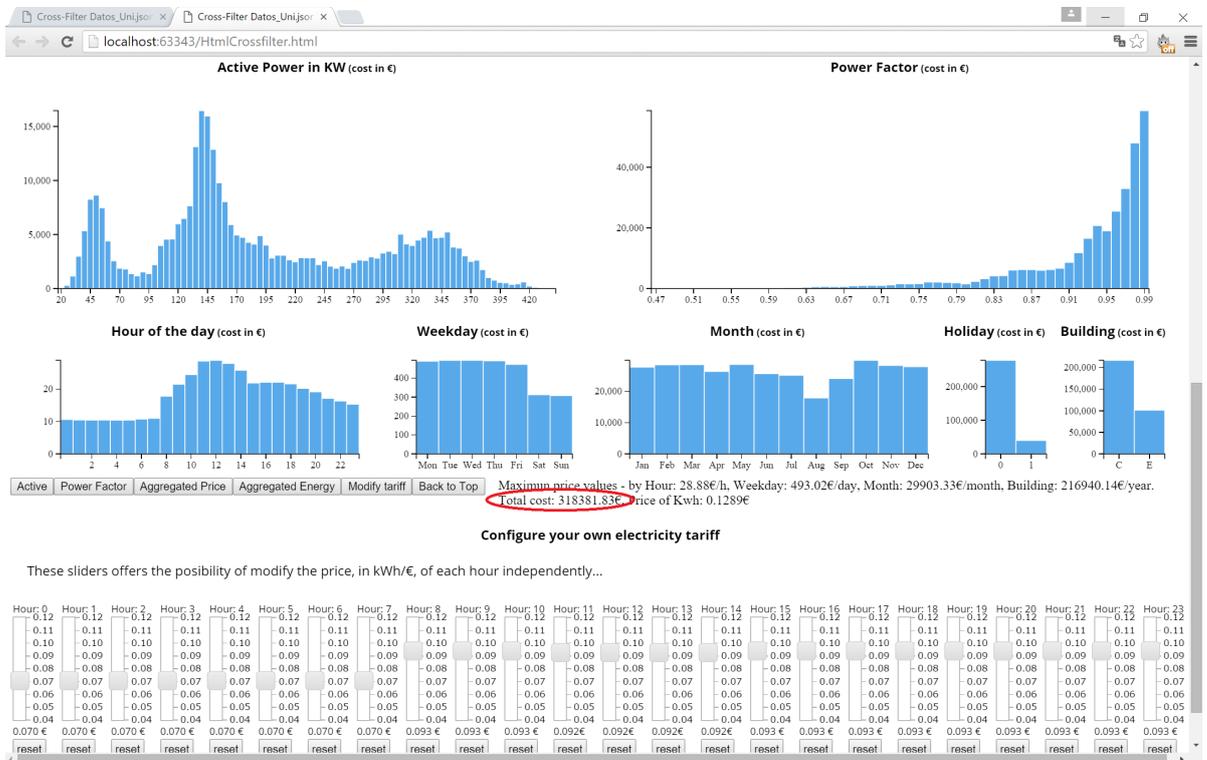


FIGURA 4.9. (a) Captura del coste total de ambas facultades, con la actual tarifa eléctrica ; (b) coste obtenido al reducir, únicamente 1 céntimo de euro, las 4 horas del tramo horario más caro. 64

## DISCUSIÓN GENERAL

**E**n este capítulo final se discutirá sobre las razones por las que las técnicas de analítica visual, aplicadas a la demanda energética, pueden, entre otras cosas, incrementar el ahorro en la factura a sus usuarios. En base a los resultados obtenidos se hablará, por tanto, de sus virtudes y se comentarán las razones por las que sólo se ha trabajado con la energía eléctrica a la hora de visualizar consumos energéticos. Ya para finalizar, se darán unas *pinceladas* de las *posibles* líneas de trabajo futuras.

### 5.1 Aportaciones

El cerebro humano es complejo, procesa la información de formas que, aún hoy, se están empezando a conocer. No es de extrañar, por tanto, que continuamente se estén buscando nuevas formas de presentar o dar a conocer una determinada información o conjunto de datos. Cada persona es distinta, tiene su manera de ver las cosas, de entenderlas y de interpretarlas, por lo que nunca habrá un sistema que se adapte perfectamente a todas ellas; lo que sí se puede decir, es que la analítica visual es una herramienta poderosa a la hora de analizar datos.

Las formas, los colores, los tamaños, están grabados en lo más profundo del ADN humano, lo que hace que se interpreten y pasen a ser asimilados por el cerebro casi instantáneamente. Si a esto se le añade *interacción* junto con *múltiples vistas coordinadas*, cada usuario puede buscar y analizar la información de la forma que mejor entiendan o necesiten. Unos buscarán ver grandes cantidades para darse cuenta de sus posibles derroches, otros quizás quieran ver cuanto están gastando de noche o cuando se van de vacaciones, otros, los de mentalidad más ecologista, quizás sean conscientes de su gasto al ver la gran cantidad de emisiones de CO<sub>2</sub> a las que equivale su

consumo eléctrico por fuentes no renovables (más adelante se profundizará en este aspecto), los más ahorradores cuando vean lo que gastan cada día (en euros), ... en definitiva, cada persona o, mejor aún, cada cerebro, tiene unos determinados "esquemas mentales" que hacen que sea indispensable el tener que facilitar unos mecanismos de interacción que les permitan, a cada una ellas, *explorar los datos a su manera*.

Este estudio, mediante la creación de esta completa herramienta de análisis, ha conseguido probar la eficacia de estas técnicas. Los diferentes *casos de estudio*, así como el *estudio de usuario* realizados en el capítulo anterior, demuestran el potencial del enfoque de analítica visual en el análisis del consumo eléctrico y, en general en la caracterización de la eficiencia energética; los primeros mostraron que se pueden explorar grandes cantidades de datos y extraer valiosa información de ellos en apenas segundos, mientras que el estudio de usuario corroboró lo ya comentado anteriormente, que cada persona tiene una determinada forma de proceder y *enfrentarse* a la información mostrada. En este último caso, los resultados fueron muy favorables ya que todas las personas fueron capaces de obtener sus propias conclusiones sobre los datos que, aunque todas correctas, no fueron siempre las mismas.

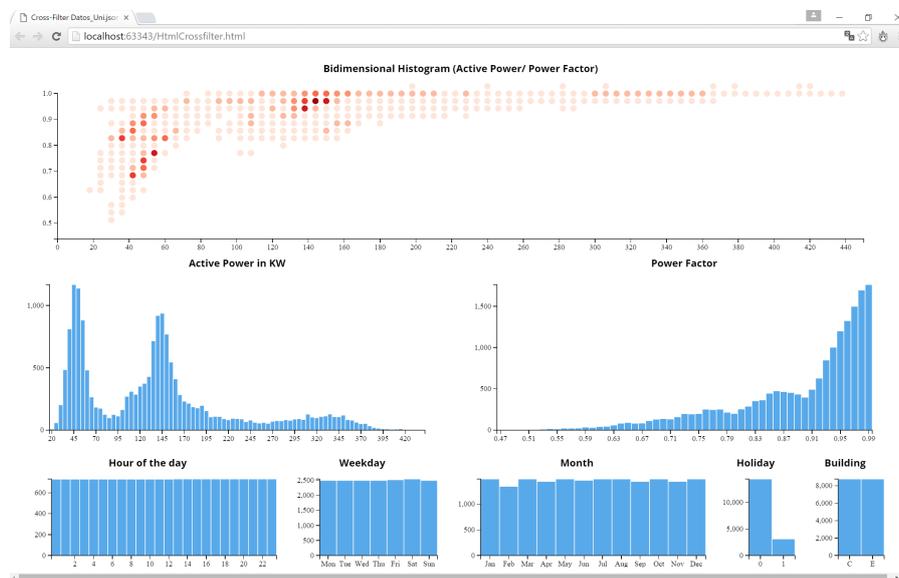


FIGURA 5.1. Captura actual de la estructura de gráficos de la aplicación. Los datos de consumos energéticos suministrados por la Universidad de Oviedo sólo contenían los eléctricos; no se dispuso de datos de consumo de gas o cualquier otra fuente de energía como pueden ser las renovables. En caso de haber dispuesto de ellos, los dos histogramas de arriba, los que muestran magnitudes eléctricas como son la *Potencia Activa* y el *Factor de Potencia*, se harían más estrechos para dejar hueco, a su derecha, a un tercero que mostraría la cantidad de energía aportada por cada fuente.

Llegados a este punto, es necesario volver a comentar que sólo se han usado datos de la demanda *eléctrica* ya que han sido los únicos suministrados. Además, la fuente de toda ella (energía eléctrica) son los combustibles fósiles. En la figura 5.1 se comenta que, de haber tenido datos de otras fuentes energéticas como gas o energías renovables, como puede ser la fotovoltaica, proporcionada por paneles solares, se habrían integrado, con muy pocas modificaciones, en la estructura de gráficos de la herramienta. Simplemente se habrían reducido los tamaños de los dos histogramas superiores, los encargados de mostrar las variables eléctricas, para dejar paso a otro que tuviera tantas barras como fuentes energéticas. De esta forma, el control de la demanda energética pasaría a un nivel superior; por todo ello, se desea remarcar que la herramienta *ha tenido eso en cuenta en su elaboración*.

## 5.2 Líneas futuras de trabajo

Se han establecido dos ramas o líneas distintas de trabajo de cara al futuro. La primera de ellas va orientada a usuarios del ámbito doméstico y la otra principalmente a empresas debido a su mayor poder de análisis y, por ende, necesidad de una mejor y más costosa arquitectura.

La de primera de ellas trata de conseguir la reducción del consumo energético proponiendo un reto a los hogares. Para ello, se pediría que, antes de comenzar la visualización, cada hogar se crearía una cuenta en la que introduciría algunas características generales de su vivienda como pueden ser los metros cuadrados o su tipo de calefacción (similar a los datos introducidos en los programas de cálculo de la eficiencia energética como el C3X, pero mucho más simplificado para hacerlo accesible a todo tipo de usuarios). Una vez introducidos, la visualización que verían sería la misma que la actual pero, sobreimpreso sobre cada histograma, se vería dibujado un gráfico de línea que mostrara los valores que la vivienda debería alcanzar para tener un consumo óptimo y eficiente. De este modo, las familias podrían ver, por ejemplo, si están gastando demasiado en calefacción en invierno y obrar en consecuencia. Como añadido, se mostraría un porcentaje de lo cerca que están de alcanzar los consumos óptimos, consiguiendo que el reto de mejorarlo sea aún mayor.

La otra línea de trabajo haría uso de *Machine Learning* para, conociendo las tendencias en el precio de la energía así como el gasto energético a lo largo de los años, poder elaborar modelos que predigan, con aceptable grado de eficacia, los consumos energéticos y gastos en euros que se tendrían de cara a unos meses o, incluso años<sup>1</sup>. Para llevar a cabo esto, se necesitaría de un ordenador con suficiente potencia que haría de *servidor* y sería el encargado de, usando librerías de *Python* como *Scikit-Learn*<sup>2</sup>, realizar los cálculos tras una petición realizada por el *cliente* (el

---

<sup>1</sup>El nivel de acierto en la predicción debería ser probado concienzudamente con gran cantidad de datos de entrenamiento y test.

<sup>2</sup>Librería de Machine Learning.

ordenador en el que se está llevando a cabo la visualización) para, posteriormente, enviarle de vuelta el nuevo conjuntos de datos objeto de la predicción para poder ser visualizados.

Ambas líneas de trabajo, con enfoques totalmente distintos, deben ser llevadas a cabo para estudiar sus posibles beneficios de cara a seguir mejorando la eficiencia en el consumo en hogares, edificios públicos o industrias; una mejora que contribuya al desarrollo sostenible tanto a nivel económico como medioambiental.



## CRONOGRAMA DEL TRABAJO

A continuación se presenta el cronograma seguido en la realización de este trabajo.

Actividades	2015															
	Septiembre				Octubre				Noviembre				Diciembre			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
INICIO																
Documentación y testeo <i>Python</i> y <i>Pandas</i>	■															
Documentación y testeo de técnicas de programación web ( <i>HTML</i> , <i>CSS</i> , <i>JavaScript</i> )	■	■	■	■												
Documentación y testeo librería <i>D3.js</i>	■	■	■	■												
Bibliografía sobre analítica visual					■	■										
Esquema del trabajo					■	■										
Elementos del trabajo					■	■										
Objetivos							■									
DESARROLLO																
Creación del esqueleto de la aplicación usando <i>D3.js</i>								■	■	■						
Introducción de <i>Crossfilter.js</i> para el uso de las múltiples vistas coordinadas										■	■					
Se implementa la función <i>reduce</i> para añadir nuevas características a la aplicación											■	■				
Optimización, adición de nuevas características y retoques visuales													■	■	■	■
CIERRE																
Redacción del borrador															■	■
Revisión y corrección del borrador															■	■
Entrega del trabajo																
Defensa																

FIGURA A.1. Cronograma del trabajo



## PRESUPUESTO DEL TRABAJO

**E**ste trabajo se ha presupuestado de forma similar a como si se tratase de un proyecto de investigación de la Universidad de Oviedo. De esta forma, los costes de gestión se han fijado en un 17% del total, siendo el beneficio, el cual iría destinado al grupo de investigación, del 8% (se ha querido mantener en un rango realista, sin hacerlo especialmente elevado). Las horas de tutorías y correcciones se han asignado a asesoría científica.

Descripción	Precio/Unidad	Cantidad	Importe
Coste personal			
Ingeniero Técnico ( €/h )	13	600	7.800 €
Material inventariable			
Amortización ordenador ( a 3 años )	140	1	140 €
Material bibliográfico	100	1	100 €
Material fungible			
Oficina (papelería, etc.)	120	1	120 €
Asesoría científica			
Doctor ingeniero industrial ( €/h )	60	20	1.200 €
Costes de gestión (17%)			1.591 €
Beneficio grupo investigación (8%)			749 €
Total			11.700 €

FIGURA B.1. Presupuesto del trabajo



## BIBLIOGRAFÍA

- [1] L. ATZORI, A. IERA, AND G. MORABITO, *The internet of things: A survey*, Computer networks, 54 (2010), pp. 2787–2805.
- [2] N. G. BELMONTE, *Javascript infovis toolkit*.  
<http://philogb.github.io/jit/>.  
[Online: accedido el 16-Diciembre-2015].
- [3] R. G. BITTLE, R. M. VALESANO, AND G. A. THALER, *Energy conservation in a large mental hospital*, Administration and Policy in Mental Health and Mental Health Services Research, 5 (1978), pp. 154–161.
- [4] M. BOSTOCK, V. OGIEVETSKY, AND J. HEER, *D<sup>3</sup> data-driven documents*, Visualization and Computer Graphics, IEEE Transactions on, 17 (2011), pp. 2301–2309.
- [5] CODEEVAL, *Most popular coding languages of 2015*, (2015).  
[Online, <http://blog.codeeval.com/codeevalblog/2015>].
- [6] S. W. COOK, *Energy conservation effects of continuous in-home feedback in all-electric homes*, Journal of Environmental Systems, 9 (1979), pp. 169–173.
- [7] J. DAVIES, *Crossfilter*.  
<http://square.github.io/crossfilter/>.  
[Online: accedido el 15-Diciembre-2015].
- [8] I. DÍAZ, M. DOMÍNGUEZ, A. A. CUADRADO, A. B. DIEZ, AND J. J. FUERTES, *Morphingprojections: Interactive visualization of electric power demand time series*, in Eurographics Conference on Visualization (EuroVis) (2012), M. Meyer and T. W. (Editors), eds., Viena (Austria), Jun 2012, pp. 121–125.
- [9] J. K. DOBSON AND J. A. GRIFFIN, *Conservation effect of immediate electricity cost feedback on residential consumption behavior*, Proceedings of the 7th ACEEE summer study on energy efficiency in buildings, 2 (1992).
- [10] D. FLANAGAN, *JavaScript: the definitive guide*, " O'Reilly Media, Inc.", 2006.

## BIBLIOGRAFÍA

---

- [11] IDAE, *Factores de emisión de CO<sub>2</sub> y coeficientes de paso a energía primaria de diferentes fuentes de energía final consumidas en el sector edificios en España*.  
[http://www.minetur.gob.es/energia/desarrollo/EficienciaEnergetica/RITE/propuestas/Documents/2014\\_03\\_03\\_Factores\\_de\\_emision\\_CO2\\_y\\_Factores\\_de\\_paso\\_Efinal\\_Eprimaria\\_V.pdf](http://www.minetur.gob.es/energia/desarrollo/EficienciaEnergetica/RITE/propuestas/Documents/2014_03_03_Factores_de_emision_CO2_y_Factores_de_paso_Efinal_Eprimaria_V.pdf).  
[Online: accedido el 16-Enero-2016].
- [12] L. MCCALLEY AND C. J. MIDDEN, *Computer based systems in household appliances: the study of eco-feedback as a tool for increasing conservation behavior*, in *Computer Human Interaction*, 1998. Proceedings. 3rd Asia Pacific, IEEE, 1998, pp. 344–349.
- [13] W. MCKINNEY, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*, O'Reilly Media, 2012.
- [14] A. R. METKE AND R. L. EKL, *Security technology for smart grid networks*, *Smart Grid*, IEEE Transactions on, 1 (2010), pp. 99–107.
- [15] L. K. MURUGESAN, R. HODA, AND Z. SALCIC, *Design criteria for visualization of energy consumption: A systematic literature review*, *Sustainable Cities and Society*, 18 (2015), pp. 1–12.
- [16] NASA, *Climate change: Vital signs of the planet*.  
<http://climate.nasa.gov/evidence/>.  
[Online: accedido el 16-Enero-2016].
- [17] P5COMMUNITY, *P5.js*.  
<http://p5js.org/>.  
[Online: accedido el 16-Diciembre-2015].
- [18] PROCESSINGCOMMUNITY, *Processing*.  
<https://processing.org/>.  
[Online: accedido el 16-Diciembre-2015].
- [19] V. RAINARDI, *Building a data warehouse: with examples in SQL Server*, John Wiley & Sons, 2008.
- [20] J. C. ROBERTS, *State of the art: Coordinated & multiple views in exploratory visualization*, in *Coordinated and Multiple Views in Exploratory Visualization*, 2007. CMV'07. Fifth International Conference on, IEEE, 2007, pp. 61–71.
- [21] C. SELIGMAN AND J. M. DARLEY, *Feedback as a means of decreasing residential energy consumption.*, *Journal of Applied Psychology*, 62 (1977), p. 363.

- 
- [22] E. R. TUFTE, *The visual display of quantitative information.*, Journal for Healthcare Quality, 7 (1985), p. 15.
- [23] W3C, *Guía breve sobre estándares web.*  
<http://www.w3c.es/Divulgacion/GuiasBreves/Estandares>.  
[Online; accedido el 15-Diciembre-2015].
- [24] M. WATTENBERG, *Visualizing the stock market*, in CHI'99 extended abstracts on Human factors in computing systems, ACM, 1999, pp. 188–189.
- [25] WIKIPEDIA, *Api* — *Wikipedia, the free encyclopedia*, 2015.  
[Online; accedido el 16-Diciembre-2015].
- [26] —, *D3.js* — *Wikipedia, the free encyclopedia*, 2015.  
[Online; accedido el 14-Diciembre-2015].
- [27] —, *Document object model* — *Wikipedia, the free encyclopedia*, 2015.  
[Online; accedido el 16-Diciembre-2015].
- [28] —, *Ecmascript* — *Wikipedia, the free encyclopedia*, 2015.  
[Online; accedido el 15-Diciembre-2015].
- [29] —, *Python* — *Wikipedia, the free encyclopedia*, 2015.  
[Online; accedido el 11-Diciembre-2015].
- [30] —, *Cambio climático* — *Wikipedia, the free encyclopedia*, 2016.  
[Online; accedido el 9-Enero-2016].
- [31] —, *Efecto invernadero* — *Wikipedia, the free encyclopedia*, 2016.  
[Online; accedido el 16-Enero-2016].
- [32] —, *Scroll* — *Wikipedia, the free encyclopedia*, 2016.  
[Online; accedido el 10-Enero-2016].

