



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería de Organización Industrial

**Análisis de Visualizaciones de Datos en D3
para una Empresa**

Autor:

Fernández Atienza, María

Tutor:

**Gonzalo Tasis, Margarita
Departamento de Informática**

Valladolid, Junio de 2018.

Resumen de TFG

En la actualidad la información es considerada como uno de los recursos más preciados, por la capacidad del ser humano de extraer conocimiento de la información disponible. La Visualización de Datos es el proceso de búsqueda, interpretación, contrastación y comparación de datos, obteniendo un conocimiento detallado de los mismos, de forma que como resultado se obtiene una información lo más comprensible posible para el usuario. En este proyecto se empleará un entorno web para llevar a cabo la implementación de gráficos a través de D3.js, una de las librerías de JavaScript más populares que permite la creación de gráficos y Visualización de Datos, así como la posibilidad de añadirlos en un navegador utilizando JavaScript, HTML5, SVG (gráficos vectoriales) y CSS, todo esto orientado a una PYME dedicada al sector industrial.

Palabras Clave

VISUALIZACIÓN DE DATOS, D3.JS, BASE DE DATOS, HTML, PHP

Abstract

Currently, information is regarded as one of the most precious resources due to the human's ability to obtain knowledge from the available information. Visualising data is the process of searching, interpreting, contrasting and comparing data, and thus we get detailed knowledge of these, in such a way that we gain some information as understandable as possible for the user. In this project a web setting will be used in order to carry out the implementation of graphics through D3.js, one of the most popular JavaScript libraries which enables us to create graphics and visualise data, as well as the possibility to display them in a browser using JavaScript, HTML5, SVG (vector-based graphics) and CSS. All this is oriented to a PYME (medium or small sized company) engaged in the industrial sector.

Keywords

DATA VISUALIZATION, D3.JS, DATABASE, HTML, PHP

Agradecimientos

En primer lugar, quería agradecer a Margarita Gonzalo Tasis por su constancia, tenacidad y apoyo durante todo el transcurso del proyecto. Sin su confianza, ayuda y paciencia no hubiera sido posible el desarrollo del presente Trabajo Fin de Grado.

También agradecer a la Universidad de Valladolid, por proporcionar los medios necesarios para la realización del Trabajo Fin de Grado, así como a la empresa colaboradora del proyecto por su colaboración para que haya sido posible la ejecución del mismo.

No me quiero olvidar de todos aquellos docentes y alumnos de la EII, que de una u otra manera han formado parte de este trabajo, estando siempre en disposición de ayudar y resolver aquellos problemas que se fueron presentando a lo largo del estudio.

Agradecer en todo momento a mis padres, a mi abuela, amigos y a esa persona especial, que han sido el ánimo y la fuerza durante todo el tiempo de estudio. Pero, sobre todo, a mi hermana que sin duda ha estado en todo momento a mi lado y ha sido mi mayor apoyo.

A todos, muchas gracias.

*“Cualquier tecnología suficientemente
avanzada es indistinguible de la
magia.”*

Arthur Charles Clarke

Índice

Contenido

Capítulo I: Introducción	19
Capítulo II: Presentación de la empresa	23
1. Descripción de la empresa y su sector.....	23
2. Motivación para la implantación de un sistema básico de información web.....	24
3. Diseño Centrado en el Usuario (DCU).....	24
Capítulo III: Introducción a la Visualización de Datos	27
1. Big Data	27
2. Data Visualization, ¿Qué es?.....	28
3. Data Visualization, ¿Para qué sirve?	29
4. Ventajas de la Visualización de Datos	30
5. Principales herramientas para la Visualización de Datos	31
Capítulo IV: D3.js en la Visualización de Datos	37
1. Creación de D3.js	37
2. Cómo funciona D3.js.....	37
2.1 Selección y vinculación.....	38
2.2 Obtención de la apariencia de los elementos de página web	39
2.3 Proceso que sufren los datos	40
3. Flujo de datos en la Visualización de Datos.....	40
3.1 Carga de datos	41
3.2 Datos de formato	42
3.3 Transformación de los datos.....	43
3.4 Medición de datos	47
3.5 Creación de gráficos	48
3.6 Estilo de presentación de datos, atributos y contenido	48
Capítulo V: Diagrama de pantallas y tecnología empleada	75
1. Diagrama de Pantallas.....	75
2. Tecnología empleada	76
3. Otra tecnología empleada	78
Capítulo VI: HTML y PHP	81
1. Introducción HTML5.....	81

2. Pantallas HTML5	82
3. Análisis pantallas HTML5	87
4. Introducción PHP	90
5. Pantallas PHP	90
6. Análisis pantallas PHP	92
Capítulo VII: Diseño de la base de datos	97
1. Modelo Conceptual de Datos	98
2. Modelo Lógico de Datos	103
3. Restricciones de Integridad	106
4. Sentencias SQL	107
Capítulo VIII: Gráfico D3.js	109
1. Selección del gráfico	109
2. Pantalla Menu_grafica.php	111
3. Análisis Pantalla Menu_grafica.php.....	111
4. Pantalla Graficas.html	112
5. Análisis Pantalla Gráficas.html.....	114
Capítulo IX: Manual de usuario.....	123
Capítulo X: Manual de instalación y planificación	131
1. Manual de instalación.....	131
2. Planificación del proyecto.....	131
Capítulo XI: Conclusiones y líneas futuras de trabajo	133
Bibliografía	135

Índice de figuras

Contenido

Figura 1. Tipo de industria	23
Figura 2. Funcionamiento D3.js	37
Figura 3. Figuras D3.js	38
Figura 4. Proceso emplea D3.js	40
Figura 5. Estructura básica empleada por D3.js.....	41
Figura 6. Carga de datos.....	41
Figura 7. Formateo de los datos.....	42
Figura 8. Escala en D3.js	44
Figura 9. Escala a color en D3.js	45
Figura 10. Binning o agrupamiento.....	46
Figura 11. Medición de datos.....	47
Figura 12. Creación de gráficos.....	48
Figura 13. Polar Clock	49
Figura 14. Floating Landmasses	49
Figura 15. Koalas to the max	49
Figura 16. Box Plots	50
Figura 17. Bubble Charts	50
Figura 18. Bullet Charts	50
Figura 19. Calendar View	51
Figura 20. Non-contiguous Cartogram	51
Figura 21. Chord Diagram.....	51
Figura 22. Dendograma	52
Figura 23. Force-Directed Graph	52
Figura 24. Circle Packing	52
Figura 25. Stacked Bars Agrupado	53
Figura 26. Stacked Bars Apilado	53
Figura 27. Streamgraph	53
Figura 28. Sunburst o Bilevel Partition	53
Figura 29. Node-Link Tree.....	54
Figura 30. Treemap tamaño	54
Figura 31. Treemap contar	54
Figura 32. Voronoi Tessellation	54
Figura 33. Hierarchical Edge Bundling.....	55
Figura 34. Voronoi Diagram	55
Figura 35. Symbol Map	55
Figura 36. Parallel Coordinates	56
Figura 37. Scatterplot Matrix.....	56
Figura 38. Hierarchical Bars	56
Figura 39. Epicyclical Gears anular	57
Figura 40. Epicyclical Gears solar	57

Figura 41. Epicyclical Gears planetas	57
Figura 42. Collision Detection.....	58
Figura 43. Collapsible Force Layout.....	58
Figura 44. Spacetime	58
Figura 45. Azimuthal Projections equidistante.....	59
Figura 46. Azimuthal Projections equalarea.....	59
Figura 47. Azimuthal Projections gnomónico	59
Figura 48. Azimuthal Projections esterográfica	59
Figura 49. Azimuthal Projections ortográfico	59
Figura 50. Choropleth	60
Figura 51. Drag and Drop Collapsible Tree Layout	60
Figura 52. Rotating Cluster Layout.....	61
Figura 53. Sankey Diagram	61
Figura 54. Confusion Matrix	61
Figura 55. Co-ocurrence Matrix cluster.....	62
Figura 56. Co-ocurrence Matrix frecuencia	62
Figura 57. Co-ocurrence Matrix nombre	62
Figura 58. Animated Bézier Curves.....	62
Figura 59. Zoomable Sunburst.....	63
Figura 60. Collatz Graph	63
Figura 61. Parallel Sets.....	64
Figura 62. World Cloud Generator.....	64
Figura 63. D3 Mobile Application	64
Figura 64. Halo	65
Figura 65. Corona Chart.....	65
Figura 66. Rounded Rectangles	65
Figura 67. Cubism.js.....	66
Figura 68. Binary Tree with Transitions	66
Figura 69. Site Concept Browser.....	66
Figura 70. Google Calendar Like Visualization.....	67
Figura 71. Bar Chart with Tooltips.....	67
Figura 72. Compare Time Series with Irregular Interval	68
Figura 73. Heat Map nombre de sonda y contraste de nombre.....	68
Figura 74. Heat Map cluster	68
Figura 75. Density and Quantile Graphs.....	68
Figura 76. Aster Plot.....	69
Figura 77. Smallest Convex Polygon	69
Figura 78. Trend Chart	70
Figura 79. Radial Boxplot.....	70
Figura 80. Visualization of a Genetic Algorithm	70
Figura 81. D3 Rectangular Area Chart.....	71
Figura 82. Cluster Purity Visualizer ángulo	71
Figura 83. Cluster Purity Visualizer longitud	71
Figura 84. Area Chart	71

Figura 85. Line Chart.....	72
Figura 86. Scatterplot	72
Figura 87. Donut Chart.....	72
Figura 88. Waterfall Chart.....	73
Figura 89. Scree Plot.....	73
Figura 90. Estructura básica documento HTML5	81
Figura 91. Principal.html	83
Figura 92. Menu.html.....	83
Figura 93. Menu_proveedores.html.....	84
Figura 94. Menu_productos.html.....	84
Figura 95. Anadir_proveedor.html.....	85
Figura 96. Eliminar_proveedor.html.....	85
Figura 97. Modificar_proveedor.html.....	86
Figura 98. Anadir_producto.html.....	86
Figura 99. Eliminar_producto.html.....	86
Figura 100. Modificar_producto.html	87
Figura 101. Código Principal.html	88
Figura 102. Código Menu.html	89
Figura 103. Código Anadir_proveedor.html.....	89
Figura 104. Proveedores.php	91
Figura 105. Productos_vendedores.php	91
Figura 106. Productos.php	92
Figura 107. Productos_precios.php	92
Figura 108. Código Anadir_Proveedor.php Parte 1	93
Figura 109. Código Anadir_Proveedor.php Parte 2	94
Figura 110. Código productos.php.....	95
Figura 111. Código usuario.php	96
Figura 112. Modelo E-R sin atributos ni cardinalidad	99
Figura 113. Modelo E-R con atributos, pero sin cardinalidad.....	99
Figura 114. Modelo E-R con cardinalidades	100
Figura 115. Modelo lógico de datos.....	106
Figura 116. Análisis Calendar View.....	109
Figura 117. Análisis Bar Chart.....	110
Figura 118. Muestra de precio y semana en cada punto del Line Chart.....	110
Figura 119. Menu_grafica.php	111
Figura 120. Código Menu_grafica.php.....	112
Figura 121. Gráfica Costillejas - Ejemplo Producto 1	113
Figura 122. Gráfica Morro - Ejemplo Producto 2	113
Figura 123. Gráfica Jamón 3D S/H - Ejemplo Producto 3.....	113
Figura 124. Código Gráfico.html Parte 1	118
Figura 125. Código Gráfico.html Parte 2	118
Figura 126. Código Gráfico.html Parte 3	119
Figura 127. Código Gráfico.html Parte 4	119
Figura 128. Código Gráfico.html Parte 5	120

Figura 129. Código Gráfico.html Parte 6	120
Figura 130. Código Gráfico.html Parte 7	121
Figura 131. Manual de usuario Página Principal	123
Figura 132. Manual de usuario Menú Principal	123
Figura 133. Manual de usuario Menú Productos	124
Figura 134. Manual de usuario Opción Visualizar Productos	124
Figura 135. Manual de usuario Opción relación entre productos y precios	125
Figura 136. Manual de usuario Opción Añadir Producto	125
Figura 137. Manual de usuario Opción Eliminar Producto	126
Figura 138. Manual de usuario Opción Modificar Producto	126
Figura 139. Manual de usuario Menú Proveedores	126
Figura 140. Manual de usuario opción Visualizar Proveedores.....	127
Figura 141. Manual de usuario Opción relación entre productos y proveedores .	127
Figura 142. Manual de usuario Opción Añadir Proveedores	128
Figura 143. Manual de usuario Opción Modificar Proveedores	128
Figura 144. Manual de usuario Opción Eliminar Proveedores	129
Figura 145. Manual de usuario Menú Gráficas	129
Figura 146. Manual de usuario Gráficas	129
Figura 147. Planificación inicial del proyecto.....	131
Figura 148. Planificación final del proyecto	132

Índice de tabla

Contenido

Tabla 1. Cardinalidades	100
Tabla 2. Proveedores	101
Tabla 3. Producto	101
Tabla 4. Semanas.....	101
Tabla 5. Año	102
Tabla 6. Usuarios.....	102
Tabla 7. Vende.....	102
Tabla 8. Tiene	103

Capítulo I

Introducción

1. Motivación y objetivos

La Visualización de Datos es un ámbito de conocimiento en continua evolución que en la actualidad se ha visto impulsado debido a la gran cantidad de información que se maneja, hasta tal punto que se podría decir que la sociedad actual vive rodeada de datos, los cuales necesitan ser analizados, interpretados y contextualizados a través de una narrativa que combina texto, imágenes y otros recursos interactivos. Sin embargo, uno de los problemas más comunes respecto a la existencia de dicho volumen de información es su falta de uso. No se trata sólo de publicar datos en la red, sino que la atención se debe centrar en cómo deben ser interpretados, de modo que se transformen en conocimiento, entrando en juego la Visualización de Datos.

La Visualización de Datos es una herramienta que permite la presentación de información para los usuarios finales. También se puede definir como el primer paso para el análisis y la proyección de datos disponibles, empleando recursos del sistema visual como procesador para detectar patrones y tendencias, ya que lo que los números no son capaces de comunicar, cuando son representados logran comunicar visualmente esta información. Este es el “poder” de la Visualización de Datos.

Las herramientas destinadas a la Visualización de Datos cubren un extenso espectro de público, desde desarrolladores de software hasta científicos de datos. Sin embargo, el presente proyecto sólo se centrará en los desarrolladores de software.

Para los desarrolladores de software existen una amplia variedad de librerías y paquetes software que contienen múltiples tipos de visualizaciones de datos. Para ellos la tarea a realizar es relativamente sencilla, ya que solo tienen que cargar estas librerías y parametrizar el tipo de gráfico que deseen crear. Es por ello que el desarrollador debe indicar los datos que quiere representar, el tipo de gráfico (líneas, barras, etc.) y la parametrización de dicho gráfico (etiquetas, escalas, etc.).

Las librerías más importantes son aquellas basadas en JavaScript, entre ellas una de las más potentes es D3.js, es por ello que el presente proyecto tratará de realizar un análisis de Visualización de Datos en D3.js para una empresa encargada de la venta de productos.

Los objetivos principales del proyecto son:

- Plantear las características principales de la Visualización de Datos, abarcando el concepto de “Data Visualization”.
- Estudiar las diferentes tecnologías que permiten la representación de datos, adentrándonos en D3.js como herramienta fundamental para la Visualización de Datos.
- Ampliar los conocimientos acerca de las bases de datos, lenguaje HTML5 y PHP.
- Desarrollar un entorno web para la PYME colaboradora del trabajo fin de grado, de forma que el usuario tenga acceso a la representación gráfica implementada con D3.js entre otras funcionalidades.
- Extraer conclusiones del trabajo fin de grado desarrollado.

2. Organización de la memoria

La memoria está organizada de la siguiente manera:

El capítulo 2 describe la empresa colaboradora en el trabajo fin de grado, así como el sector industrial al que pertenece. También se explica la motivación para la implantación de un sistema básico de información web y cuál es el proceso de un diseño centrado en el usuario.

El capítulo 3 presenta los fundamentos de la Visualización de Datos, abarcando sus características y conceptos más importantes, así como las ventajas que proporciona al usuario y las principales herramientas para la Visualización de Datos.

El capítulo 4 analiza todos los aspectos de D3.js abarcando sus orígenes y funcionamiento, así como el flujo de datos en la Visualización de Datos, incluyendo los diferentes tipos de gráficos que ofrece esta herramienta de visualización.

El capítulo 5 presenta los diagramas de pantalla como una secuencia de procesos de software basados en usuario, así como la tecnología empleada para poder realizar el proyecto.

El capítulo 6 muestra la fase de diseño en su totalidad, es decir, las pantallas y ficheros creados para la ejecución de la página web, tanto en HTML como en PHP, analizando la funcionalidad de cada uno de ellos.

El capítulo 7 analiza la base de datos que ha sido necesaria crear para poder llevar a cabo el proyecto mediante el modelo conceptual de datos, modelo lógico de datos, las restricciones de integridad y las sentencias SQL utilizadas.

El capítulo 8 abarca el gráfico creado con la herramienta de Visualización de Datos D3.js, con las diferentes pantallas y ficheros creados para su ejecución, tanto en HTML como en PHP, y todo ello, a través del empleo de lenguaje JavaScript.

El capítulo 9 presenta el manual de usuario cuyo objetivo es proporcionar una guía al usuario, permitiendo que pueda hacer uso correcto de la página web, así como facilitarle el uso de la misma.

El capítulo 10 muestra el manual de instalación, que incluye todos los programas que han sido necesarios para el desarrollo del trabajo, así como la planificación llevada a cabo en el presente proyecto.

El capítulo 11 muestra las conclusiones obtenidas en la realización del trabajo fin de grado.

Capítulo II

Presentación de la empresa

1. Descripción de la empresa y su sector

La empresa que nos compete en el presente proyecto recibe el nombre de “Mafrilasa” y se enmarca dentro del “sector secundario o industrial”.

“Mafrilasa” es una empresa creada en 1991 y localizada en Laguna de Duero, en Valladolid. Su actividad está destinada al procesado y conservación de productos cárnicos, lo que la sumerge dentro del sector secundario, y presenta un rango de ventas anual entre 3 y 6 millones de euros.

El sector secundario de una economía es el conjunto de actividades a través de las cuales las materias primas son transformadas en bienes manufacturados de consumo. Este concepto suele ir ligado al de industria o actividad manufacturera, la cual se encarga de la producción de bienes elaborados o semielaborados a partir de materias primas, mediante procesos de transformación que requieren la utilización de capital y trabajo, es por esta razón, que también es conocido como sector industrial. Por lo tanto, para llevar a cabo su actividad este sector suele depender en la mayoría de los casos del sector primario como suministrador de materias primas y del sector terciario para financiarse y contratar servicios auxiliares [1].

Dentro del sector secundario se puede destacar cuatro grandes grupos de actividades que engloban a la amplia mayoría de las funciones desarrolladas por dicho sector, estas son: Minería, Construcción, Energía e Industria. En este caso, debido a la empresa que compete el proyecto, se va a centrar la atención en el ámbito de la Industria.



Figura 1. Tipo de industria

En el sector secundario de la industria se incluyen las actividades manufactureras, que pueden clasificarse en talleres o fábricas, y a su vez reclasificarse en industrias ligeras (aquellas destinadas a la producción de bienes de consumo) o pesadas (dedicadas a la obtención de bienes de equipo o de capital). La empresa colaboradora en el presente proyecto pertenece a la industria ligera que de nuevo puede ser catalogada como se puede ver en la figura 1.

Por tanto, en este caso se estaría hablando de una empresa que pertenece al sector secundario de la industria ligera de alimentación.

2. Motivación para la implantación de un sistema básico de información web.

A continuación, se expondrán las diferentes razones y motivaciones por las cuales la empresa ha decidido llevar a cabo la implantación de un sistema básico de información web. Entre ellas se puede encontrar:

- Aumento de la productividad y eficacia en la gestión de información transversal, dado que se accede a la información con mayor rapidez.
- Evitar la duplicidad documental y nichos de información que pueden desencadenar muchos problemas.
- Obtención de un entorno en el que simplifica las metodologías de trabajo empleadas.
- Necesidad de evolución tecnológica.

Las razones que han sido mencionadas a continuación, son suficientes para abordar la creación e implantación de un sistema básico de información web.

3. Diseño Centrado en el Usuario (DCU).

El diseño centrado en el usuario (DCU) consiste en enfocar el diseño de un producto con la información necesaria que vayan a emplear las personas a las que va dirigido. La funcionalidad que aportan los objetos es mucho mayor, dado que se ajustan a las necesidades que los usuarios están buscando.

Para entender el proceso son necesarios tres términos:

- El modelo conceptual: Ofrecido por el diseñador del sistema.
- Interfaz: La imagen que el sistema presenta al usuario.
- El modelo mental: Desarrollado por el usuario a partir de la imagen.

De forma que el objetivo principal es el diseño de un sitio web atendiendo a la estructura y necesidades de la empresa.

Entre las características principales del diseño centrado en el usuario podemos encontrar:

- Especificar el contexto del uso: Identificar las personas que van a utilizar el objeto, por qué lo utilizaran y bajo qué circunstancias.
- Incluir los requisitos: Aquellos requisitos que buscan tanto las empresas como los usuarios para conseguir las expectativas establecidas.
- Desarrollar soluciones: Este proceso se debe llevar a cabo en las distintas fases de desarrollo del producto.

De manera, que como se ha visto el diseño centrado en el usuario tiene como objetivo la satisfacción de todas las necesidades de aquellos que visitan la web [2].

Capítulo III

Introducción a la Visualización de Datos

1. Big Data

Big Data es el conjunto de datos o combinaciones de conjuntos de datos cuyo tamaño (volumen), valor, complejidad (variabilidad) y velocidad de crecimiento dificultan su captura, gestión, procesamiento o análisis a través de herramientas y tecnologías convencionales. El volumen, valor, variabilidad y velocidad de crecimiento reciben el nombre de las “4V”, y a continuación, se irán describiendo una a una.

El volumen hace referencia a la cantidad de datos, de forma que cuanto mayor sea más datos hay. Big Data requiere el procesamiento de grandes volúmenes de datos de baja densidad, datos de valores desconocidos, como pueden ser tráfico de red, aplicaciones móviles, etc. La principal tarea de Big Data es llevar a cabo la conversión de estos datos de valores desconocidos a información útil. Además, en función del tipo de empresa se puede estar hablando desde decenas de terabytes a cientos de petabytes. La velocidad es la característica más asociada al Big Data.

La variedad se refiere a la gran cantidad de tipos diferentes de datos que se puede encontrar. Existen nuevos tipos de datos “no estructurados”, entre los que se puede encontrar textos, audios o videos, los cuales requieren un procesamiento adicional para la obtención de su significado y metadatos de apoyo. Sin embargo, una vez que los datos no estructurados son comprendidos, presentan los mismos requisitos que los datos estructurados, como el alineamiento, privacidad, etc. Además, no se debe olvidar que hay una mayor complejidad cuando los datos de origen conocido cambian sin previo aviso.

La velocidad es el ritmo al que los datos son recibidos, así como el ritmo con el que se lleva a cabo alguna acción, es decir, son los datos en movimiento por las constantes interconexiones realizadas. Algunas aplicaciones de Internet poseen ramificaciones de estado y seguridad, como puede ser la detección de un fraude en una transacción bancaria, y que por tanto requieren acciones y evaluaciones en tiempo real. También existen productos inteligentes preparados para Internet, que funcionan en tiempo real como las aplicaciones de comercio electrónico para consumidores que intentan llevar a cabo una combinación de la ubicación del dispositivo móvil con sus preferencias personales para realizar ofertas de marketing en función de la hora.

El valor hace referencia al valor intrínseco que poseen todos los datos y que debe ser descubierto. En la actualidad existe una amplia gama de técnicas cuantitativas y de investigación para extraer el valor de los datos, como la aplicación de una

oferta por la ubicación o la identificación de una parte del equipo que se encuentre a punto de dar un fallo. La innovación tecnológica ha permitido que el coste de la computación y del almacenamiento de datos haya disminuido exponencialmente, de forma que se puede realizar un análisis estadístico de una gran cantidad haciendo uso de todo el conjunto de datos, en vez de una única muestra, lo que permite llevar a cabo la toma de decisiones de un modo preciso y exacto. Sin embargo, para la obtención del valor es necesaria la participación de analistas, usuarios y ejecutivos con grandes conocimientos.

En la actualidad se habla de otras 3V más, como es la veracidad, viabilidad y visualización de datos.

La veracidad es la incertidumbre de datos, es decir, el grado de fiabilidad de la información recibida. Es necesario invertir tiempo para conseguir datos de calidad, a través de la aplicación de soluciones y métodos que puedan eliminar datos imprevisibles que puedan surgir como datos económicos, comportamientos de consumidores que puedan influir en las decisiones de compra, etc.

La viabilidad es la capacidad que tienen las diferentes empresas de hacer un uso eficaz del gran volumen de datos que se maneja. La viabilidad la encontramos relacionada con la inteligencia empresarial, pues esta es asociada con la capacidad de innovación y el uso de tecnologías. Una empresa inteligente es aquella que analiza, selecciona y monitoriza la información con el objetivo de conocer a la perfección el mercado, a sus clientes y a su vez ser capaces de diseñar estrategias eficaces. Sin embargo, es necesario filtrar toda esta información de la que se dispone, así como llevar a cabo una selección de factores, que sean capaces de predecir los resultados que más interesan a las empresas. Se puede decir que el éxito se encuentra en descubrir las relaciones entre variables ocultas.

Por último, la Visualización de Datos es el modo en que los datos son presentados. Una vez que los datos son procesados, es necesario representarlos de forma que sean legibles y accesibles, para poder encontrar diferentes patrones o claves ocultas. Para llevar a cabo la Visualización de Datos, en la actualidad existen diferentes herramientas de visualización que permite una comprensión de los datos gráficamente y en perspectiva contextual.

El verdadero reto de Big Data se encuentra en el factor humano, es decir, reconocer patrones, predecir comportamientos, realizar suposiciones fundamentadas y aprender a formular preguntas adecuadas [3].

2. Data Visualization, ¿Qué es?

En la actualidad uno de los sectores que más está evolucionando en el software científico es la Visualización de Datos. Se puede definir la Data Visualization como el proceso de búsqueda, interpretación, contrastación y comparación de datos de

forma que permite obtener un conocimiento detallado de los mismos, teniendo como objetivo la comunicación del significado de la forma más adecuada para cada usuario.

El término de Data Visualization apareció con el nacimiento de la web 2.0, debido a la gran cantidad de datos cuya complejidad en su búsqueda e interpretación, da lugar a la necesidad de un mecanismo que permitiese el entendimiento y comprensión de toda esta información de un modo sencillo. Sin embargo, el requisito principal en el proceso es saber comunicar la información obtenida, de forma que se adquieran los conocimientos necesarios para poder realizar una comprensión de los datos. Para ello, el usuario creador de la visualización debe tener muy clara la información que desea comunicar, para que resulte sencillo transmitírsela a otros usuarios a través de las visualizaciones.

No se debe olvidar que la Visualización de Datos es una herramienta que ayuda al análisis y no es un sustituto de la habilidad analítica. Tampoco es un sustituto de las estadísticas, es más, el gráfico puede resaltar las diferencias o correlaciones entre los puntos de datos.

Tampoco se debe olvidar que disponer de una gran herramienta de Visualización de Datos no es suficiente, además es necesario disponer de conocimientos de negocio, estadística, composición gráfica e inteligencia emocional [4].

3. Data Visualization, ¿Para qué sirve?

Existen un gran número de motivos por los que resulta interesante no sólo realizar la representación visual de datos, sino además hacerlo de un modo que el público objetivo lo entienda rápidamente, como por ejemplo la búsqueda de financiación o la presentación de la viabilidad de un proyecto entre otros.

Data Visualization permite la extracción de visualizaciones de Big Data de un modo rápido, siendo a su vez, una herramienta de comunicación entre la dirección de una compañía y los analistas.

El objetivo principal de la Visualización de Datos es comunicar información o ideas complejas de un modo preciso, claro y eficiente, de forma que permita a los usuarios llevar a cabo análisis sobre datos y evidencias.

Si nos preguntásemos: ¿Quién puede utilizar y sacar partido de la Visualización de Datos? La respuesta sería desde grandes a pequeñas empresas, entes educativos, y por supuesto, sin olvidar a cualquier trabajador involucrado en cualquier tipo de proyecto.

Para llevar a cabo la Visualización de Datos son necesarios 3 momentos, en primer lugar, la recopilación de información o toma de datos, en segundo lugar, la organización de los datos, y por último, la estética y diferenciación.

La recopilación de los datos que se desean representar es tan importante, como la propia representación gráfica, sin olvidar que la veracidad de los datos es uno de los factores más importantes. Sin embargo, también se puede realizar esta representación gráfica utilizando datos puramente especulativos, en función de los objetivos que se persigan. Los diferentes datos se pueden encontrar y reunir gracias a la suma de resultados de una actividad, formularios, encuestas, monitorización y analítica entre otros.

La organización de los datos requiere una clasificación de los distintos datos de los que se dispone haciendo uso por ejemplo de mapas mentales, los cuales permiten organizar, seleccionar, valorar y eliminar cualquier información. Sin embargo, existen otros métodos para llevar a cabo la organización de información, como puede ser bocetos, flechas, iconografías, etc.

Por último, la estética y la diferenciación. Para realizar una representación gráfica de datos, se pueden utilizar distintos recursos entre los que se pueden encontrar gráficos de barras, mapas mentales, diagramas, etc. Es muy importante dotar la información de un componente o aspecto visual agradable para una fácil y rápida asimilación de los usuarios. Además, la diferenciación permite destacar un elemento entre el resto. El uso del color desempeña un papel muy importante en función del medio en el que se reproduzca, ya que ayuda a resaltar, distinguir o dar mayor importancia a aquellos aspectos que interesen.

4. Ventajas de la Visualización de Datos

La Visualización de Datos presenta una gran serie de beneficios para aquellas empresas que lo empleen en su trabajo. Entre las ventajas del uso de gráficos, diagramas o mapas interactivos se puede encontrar:

Mayor visibilidad sobre los procesos de negocio y la actividad empresarial: Los usuarios pueden ver las conexiones, a medida que se produzcan, entre las condiciones operativas y el rendimiento del negocio. Estas correlaciones resultan muy importantes para un entorno empresarial tan competitivo como los que encontramos en la actualidad, debido a la introducción de ajustes efectivos sin consecuencias de las latencias (suma de retardos temporales dentro de una red).

Mejor comprensión de datos: La Visualización de Datos permite a los distintos usuarios recibir grandes cantidades de información acerca de las condiciones operativas y comerciales de un modo ágil. Además, el contacto con el conocimiento conlleva un impulso al descubrimiento, dado que la visualización proporciona nuevas formas y métodos de llevar a cabo la interpretación de datos y de esta manera permite realizar un planteamiento de cuestiones que de otro modo jamás hubiesen sido planteadas.

Minimización de errores: La interacción directa con datos fomenta la aparición de errores. Estos fallos son del tipo humano, los cuales aparecen durante la fase de

negocio. Cuando, por el contrario, se tiene acceso a la información de forma visual, no existe riesgo de confusiones. Además, la Visualización de Datos permite la profundización en cada dato, lo que lleva a algo tan importante como el acceso sin límites a un conocimiento actualizable.

Identificación de tendencias: En los casos en los que se trabaja con grandes volúmenes de datos, la Visualización de Datos se presenta como el mejor aliado del conocimiento. Big Data permite a los líderes del negocio innovar y, por tanto, emplear nuevas ideas que les facilite un aumento de los ingresos. Las oportunidades de negocio aumentan cuando se comprenden los cambios de las condiciones del mercado y el comportamiento de los clientes de un modo ágil.

Acceso rápido a estadísticas de negocios relevantes: En muchas ocasiones el éxito de un negocio reside en el tiempo. Solo se necesitan unos minutos para tomar una decisión acertada, que, si hubiera sufrido un retraso por causa del propio proceso, ya no sería ni la mitad de efectiva. Con esto se desea explicar que aquellos gerentes de negocio que emplean herramientas de descubrimiento de datos visuales tienen más probabilidades de encontrar información oportuna que les permita llevar a cabo un negocio exitoso, que aquellos gerentes de negocio que únicamente confían en informes y cuadros de mando administrados. Según un estudio de *Aberdeen* esta probabilidad se encuentra en torno al 28% a favor de aquellos agentes que utilizan herramientas de descubrimiento de datos visuales [5].

5. Principales herramientas para la Visualización de Datos

La posesión de una buena información con datos fiables es el primer paso para atraer la atención de los distintos usuarios. Hoy en día en la red podemos encontrar multitud de herramientas que permiten visualizar datos de una forma clara y mucho más efectiva que si nos limitásemos a la realización de una presentación tradicional.

Cada herramienta posee sus propias características, sin embargo, la mayoría de ellas funcionan bajo la lógica de importar datos, elegir las opciones de visualización y finalmente publicarla.

Actualmente las herramientas más conocidas se aseguran de que puedan ser utilizadas sin tener conocimientos de programación, y además, que los gráficos que proporcionan se encuentren optimizados para móviles, pudiendo ser compartidas en el momento las visualizaciones creadas a través de las redes sociales. Asimismo, este tipo de herramientas ofrecen versiones gratuitas, buenas prácticas a la hora de usarlas y comunidades de usuarios para compartir consejos.

Entre las herramientas más populares y recomendadas en la Visualización de Datos encontramos:

Tableau Software

Tableau Software es una herramienta que destaca por su facilidad para integrar diversos tipos de fuentes de datos y para crear dashboard que faciliten la toma de decisiones, a partir de esta información. Dicho de otro modo, es un software de Business Intelligence, que permite realizar análisis de datos, así como mejorar y acelerar la toma de decisiones de la empresa.

Entre las principales características de Tableau Software se pueden encontrar:

- Rapidez y facilidad de uso: Acceso veloz a millones de datos, es decir, permite acceder a más datos con menos hardware.
- Cuadros de mando brillantes: Permite la combinación de varios análisis en un único informe donde se inserten páginas webs y documentos para una mejor comprensión.
- Conexión directa: No requiere programación.
- Servicio gratuito.
- Permite a los usuarios explorar datos en línea, así como compartir sus visualizaciones.

QlikView

QlikView es una herramienta de Business Intelligence que permite la recolección de datos de diferentes orígenes, basados en ERP, CRM, bases de datos SQL, datos de Excel, entre otros. Así como su empleo a gusto del usuario para hacer más sencillo su manejo y presentación de forma visual.

Entre las principales características de QlikView se pueden encontrar:

- Se encuentra disponible en varias versiones, de forma que permite su uso desde diferentes plataformas como son Windows, Mac, Linux e incluso desde dispositivos móviles como iPad, iPhone o Smartphone con Android.
- Dispone de su propio lenguaje de modelado de datos, lo que le permite que trabaje desconectado de los orígenes de los datos, ya que los datos se integran con el propio cuadro de mandos.
- Posee búsqueda asociativa, es decir, permite a los usuarios la búsqueda e interacción de sus datos de un modo asociativo. Dicho de otro modo, los usuarios pueden ver al instante las conexiones y relaciones entre los datos residentes de distintas aplicaciones, sistemas, organizaciones y regiones.
- Permite la interacción en tiempo real con los datos.

Datawrapper

Datawrapper es una herramienta de visualización en línea que permite a los distintos usuarios que lo utilizan, la creación de gráficos y mapas de un modo

sencillo y rápido. Es decir, es una herramienta de código abierto que permite la creación de gráficos simples, correctos e integrables fácilmente.

Entre las principales características de Datawrapper se pueden encontrar:

- Software de código abierto: Se tiene acceso al código fuente para poder modificarlo, corregirlo o añadir más prestaciones.
- Facilidad para la generación de gráficos y mapas.
- Permite la incrustación en HTML `<iframe>`: este término permite la inserción de un documento HTML dentro de un documento HTML principal.
- Es totalmente modular.

CartoDB

CartoDB es una herramienta formada por una plataforma en la nube de código abierto que proporciona inteligencia de localización y Visualización de Datos a través de mapas. Esta herramienta presenta dos partes fundamentales, la primera es un editor desde la web que importa datos, crea y publica los mapas temáticos; y la segunda es una plataforma que permite utilizar varias aplicaciones para la construcción de conjuntos de datos geoespaciales más complejos y por tanto permite la creación aplicaciones propias.

Entre las principales características de CartoDB se pueden encontrar:

- Es un software libre: Lo que implica que puede ser copiado, estudiado, modificado, utilizado libremente con cualquier fin o redistribuido con o sin cambios o mejoras.
- Presenta una gran capacidad: Permite trabajar con gran cantidad de datos, de forma que a medida que aumenta la cuenta aumenta la capacidad.
- Puede ser utilizado con cualquier lenguaje de programación a través de sus APIs, es decir, cualquier lenguaje que pueda realizar conexiones HTTP puede utilizarlo.

Google Maps

Google Maps es un conjunto de APIs que permiten la superposición de datos sobre un mapa de Google Maps personalizado, de forma que el usuario puede crear aplicaciones web y móviles con imágenes satélite, perfiles de elevación, mapas con estilos, demografía y una amplia base de datos de ubicaciones, con la cobertura global más precisa del mundo.

Entre las principales características de Google Maps se pueden encontrar:

- Mejora la toma de decisiones: Esta API ofrece una herramienta de análisis de forma que se puede visualizar en todo momento como el resto de los usuarios interactúan con mapas creados por otro usuario.
- Optimiza recursos: A través de la interfaz de Google Maps, se pueden crear aplicaciones para aprovechar el seguimiento de activos, alertas basadas en restricciones geográficas y localizaciones geográficas de dispositivos.
- Involucra a los usuarios: Permite una gran interacción con otros usuarios, a través de la inclusión de mapas en el sitio web y aplicaciones.
- Concede prioridad a los dispositivos móviles, además destaca por su sencillez y por la calidad de sus imágenes.

Many Eyes

Many Eyes es una herramienta de Visualización de Datos que ha puesto a disposición de los usuarios la empresa IBM (International Business Machines), la cual permite que las series de datos estadísticos puedan ser representados o en forma numérica o a través de una representación gráfica.

Entre las principales características de Many Eyes se pueden encontrar:

- Permite a los usuarios compartir las visualizaciones en las redes sociales.
- La herramienta es de uso público, es decir, que todos aquellos datos y visualizaciones que se empleen estarán a disposición del resto de usuarios.
- Permite que exista conversación alrededor de una visualización y que por tanto otros usuarios aporten su punto de vista a otras visualizaciones y propongan otros enfoques.
- Servicio gratuito.

Google Fusion Tables

Google Fusion Tables es una API de Google dedicada a la gestión de datos, donde los datos son almacenados en distintas tablas que los usuarios de este tipo de herramienta pueden ver y descargar. El servicio web proporciona un medio para la Visualización de Datos a través de gráficos circulares, gráficos de barras, diagramas de dispersión, líneas de tiempo e incluso mapas geográficos basados en Google Maps.

Entre las principales características de Google Fusion Tables se pueden encontrar:

- Permite la combinación de los datos de la información que se encuentre en línea.

- Los usuarios pueden cargar sus propias tablas desde hojas de cálculo o archivos CSV.
- Habilita el uso de tablas sin la creación de cientos de copias.
- Servicio gratuito.

Infogr.am

Infogr.am es una herramienta online que permite la creación de infografías, es decir, la representación visual empleada para explicar un concepto, una idea o datos complejos de un modo claro y atractivo para los diferentes usuarios, a través de gráficos, mapas, símbolos, diagramas o líneas de tiempo.

Entre las principales características de Infogr.am se pueden encontrar:

- Facilidad de uso.
- No requiere la necesidad de instalar el programa en el ordenador, es una herramienta online.
- Permite a los usuarios compartir las visualizaciones en las redes sociales.
- Servicio gratuito.

Gephi

Gephi es un software open-source desarrollado en Java que permite el análisis y la visualización de grandes gráficos de red en tiempo real, para ello emplea un motor de renderizado 3D. Esta herramienta permite explorar, analizar, filtrar, clusterizar, manipular y exportar diversos tipos de datos.

Entre las principales características de Gephi se pueden encontrar:

- Visualización en tiempo real.
- Presenta una arquitectura modular, es decir, una arquitectura caracterizada por su versatilidad, tecnología y rapidez.
- Software de código abierto: Se tiene acceso al código fuente para poder modificarlo, corregirlo o añadir más prestaciones.
- Servicio gratuito.

Todas estas herramientas además de por su popularidad destacan por la gran variedad de formas que permiten para poder representar los distintos datos, así como su gran utilidad en la actualidad. Sin embargo, en el presente trabajo nos centraremos en una herramienta destinada a la Visualización de Datos que se encuentra en pleno auge en la actualidad [6].

Capítulo IV

D3.js en la Visualización de Datos

D3.js (también llamado Data Driven Documents o simplemente D3) es una de las librerías de JavaScript más populares que permite la creación de gráficos y Visualización de Datos (visualizaciones interactivas), así como la posibilidad de añadirlas en un navegador utilizando JavaScript, HTML5, SVG (gráficos vectoriales) y CSS. Esta librería es sucesora de la librería Protovis y a diferencia de muchas otras librerías, D3.js permite tener el control completo sobre el resultado final.

1. Creación de D3.js

El creador de D3.js, Mike Bostock, antes del comienzo de este proyecto, ayudó al desarrollo de la biblioteca de Visualización de Datos, Protovis, así como también Polymaps, una biblioteca de JavaScript con capacidad de mapeo vectorial y de mosaicos de forma ligera. Ambos desarrollos, condujeron a la creación de D3.js, que se centra en estándares y navegadores modernos. Bostock la evalúa de la siguiente manera: *“D3.js evita la representación de la propiedad y ofrece una flexibilidad extraordinaria, exponiendo todas las capacidades de los estándares web como CSS3, HTML5 y SVG”*.

2. Cómo funciona D3.js

Para poder entender el funcionamiento de D3.js, se visualizará la figura 2 en la que se puede ver un mapa de cómo usar D3 para el procesamiento y la representación de los datos, así como agregar interactividad u optimizar la Visualización de Datos.

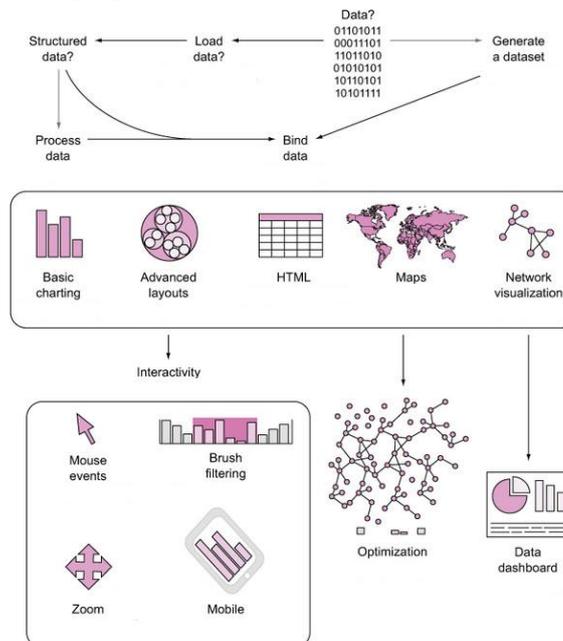


Figura 2. Funcionamiento D3.js

Se puede pensar que la Visualización de Datos queda limitada únicamente a gráficos de líneas, circulares y variedad de métodos gráficos, sin embargo, es una equivocación, puesto que uno de los puntos fuertes que presenta D3.js es que permite la creación de vectores gráficos para gráficos tradicionales, así como también la creación de visualizaciones geoespaciales y de red, incluyendo elementos HTML tradicionales como tablas, listas o párrafos.

En la figura 3 se puede ver una muestra general con algunos de las posibilidades que ofrece este software y que más adelante se analizará detalladamente.

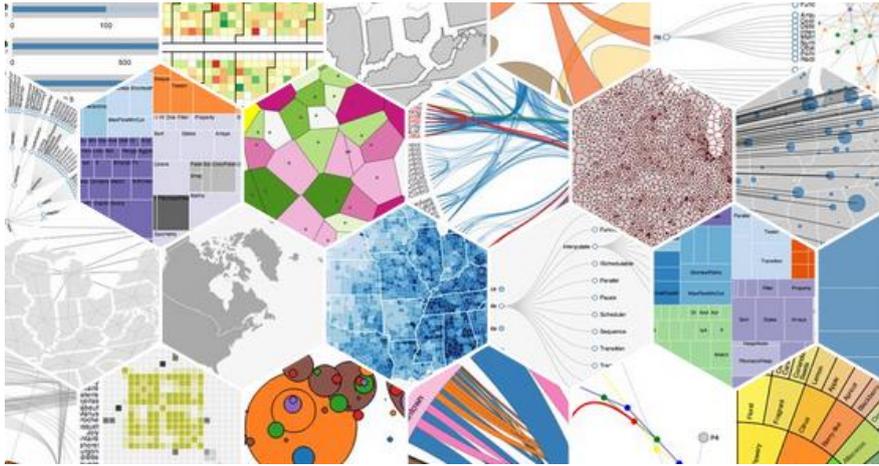


Figura 3. Figuras D3.js

Como ya se ha dicho anteriormente, uno de los puntos fuertes de D3.js es la capacidad de crear gráficos variados, sin embargo, lo más importante para el desarrollo web es la incorporación de un alto nivel de interactividad, es por ello que cada elemento de cada gráfico de D3.js presenta esta interactividad.

La decisión de Bostock de crear una biblioteca con la capacidad de representar mapas con tanta facilidad como gráficos, implica que el desarrollador no tiene que tratar de entender la sintaxis de una biblioteca para mapas, otra para contenido de texto dinámico y una última para la Visualización de Datos. Además, el código para la ejecución de una red interactiva se encuentra muy cerca de JavaScript puro y también es muy parecido al código que presenta los puntos de interés dinámico en un mapa D3.js, pues no sólo los métodos son los mismos, sino que los mismos datos pueden ser los mismos, formulados de una manera para listas, párrafos y tramos, y de otra para la representación geoespacial.

2.1 Selección y vinculación

Se analizarán fragmentos de código que pueden ser empleados en el navegador para realizar cambios en la apariencia gráfica de los elementos en su sitio web.

En el caso de que se tuviese un conjunto de datos, como por ejemplo los precios de un conjunto de artículos, y un conjunto de elementos de la página web, y se quiere representar dicho conjunto de datos, con texto o por formato de tamaño y color, en este caso, una opción es realizar la selección de todos ellos juntos y efectuar distintas acciones sobre los elementos seleccionados, como moverlos, cambiar el color o actualizar los valores de los datos.

Ejemplo de selección sin ningún dato:

```
d3.selectAll("circle.a").style("fill", "red").attr("cx", 100);
```

Esto implica que se coge cada círculo de la página web con la clase de “a”, lo transforma a color rojo, y lo mueve de forma que su centro esté a 100 píxeles a la derecha del lado izquierdo del <svg> lienzo.

```
d3.selectAll("div").style("background", "red").attr("class", "b");
```

En este otro ejemplo, el código convierte cada div de la página web a color rojo y transforma su clase a “b”.

Si se analiza detalladamente el primer código, *d3.selectAll()*, hace referencia a la selección anteriormente mencionada, aunque también se puede realizar la selección haciendo uso de *d3.select()*. Las selecciones son un grupo de uno o más elementos de página web que pueden asociarse con un conjunto de datos.

```
d3.selectAll("div.market").data([1,5,11,3])
```

Este código, vincula los elementos en la matriz [1,5,11,3] a <div> elementos con la clase de “market”. Esta asociación es conocida como “datos vinculantes”, los cuales permiten pensar en una selección como un conjunto de elementos de página web y un conjunto de datos asociados.

2.2 Obtención de la apariencia de los elementos de página web

Una vez realizada la selección, D3.js permite la modificación de la apariencia de los elementos de una página web para mostrar las diferencias en los datos a partir de datos encuadrados. Esto implica, que puede hacer que la longitud de una línea sea igual al valor de los datos, o el cambio de un color a otro particular que corresponda a una clase de datos.

También se puede realizar la modificación de la apariencia de los elementos utilizando selecciones para hacer referencia a datos vinculados a un elemento de la selección, esto es debido, a que D3.js itera a través de los elementos en su selección y realiza el mismo proceso empleando datos encuadrados, obteniendo como resultado “gráficos efectos”.

2.3 Proceso que sufren los datos

A continuación, se analizará el proceso que emplea D3.js, para dar respuesta a la interacción de un usuario, con el fin de crear, eliminar o cambiar elementos del DOM (modelo de objetos del documento).

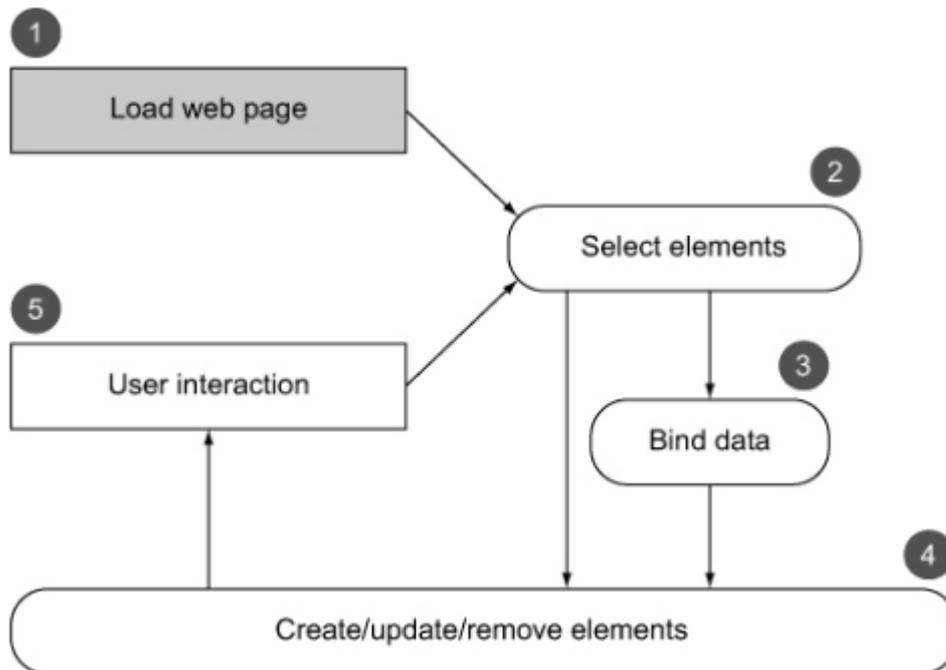


Figura 4. Proceso emplea D3.js

El primer paso de toda página web que emplea D3.js consiste en la **construcción**, de forma que la página web carga diferentes datos, estilos y contenido. A continuación, le sigue la **visualización inicial** empleando selecciones D3.js de elementos HTML. Posteriormente, por lo general se suele producir el **enlace de los datos**, aunque no siempre es necesario, para llegar al cuarto paso, el cual permite la **modificación** de la estructura y apariencia de la página web. Finalmente, el último paso, en el cual los cambios en la estructura solicitan la **interacción** del usuario, lo que conduce a nuevas selecciones con o sin visualizaciones de datos. Cabe destacar, que el primer paso solo sucede una única vez (cuando el usuario carga la página web), sin embargo, los cuatro pasos restantes pueden ocurrir varias veces, en función de la interacción del usuario.

3. Flujo de datos en la Visualización de Datos

Los datos que van a ser empleados por D3.js al igual que en muchas otras herramientas destinadas a la Visualización de Datos, provendrán de una API o de un archivo externo, que tendrá que ser cargado, formateado y transformado antes de empezar a crear elementos web basados en esos datos.

Las estructuras básicas que son empleadas en D3.js son: **cargar** datos desde una fuente externa, **formatear** dichos datos y **crear** representaciones gráficas con estos datos, como bien puede verse en la figura 4.

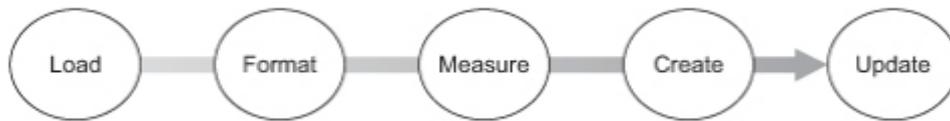


Figura 5. Estructura básica empleada por D3.js

3.1 Carga de datos

Como ya se ha dicho anteriormente, los datos que van a ser empleados necesitan ser formateados antes de ser utilizados, para ello independientemente de la fuente de datos, generalmente se formateará como archivos de datos de documento único en formato XML, CSV o JSON.

Una de las principales diferencias entre estos formatos, se encuentra en la forma en que se modelan los datos. Los formatos JSON y XML proporcionan la capacidad de codificar relaciones anidadas, a diferencia de otros formatos delimitados como CSV que no lo hacen. Otra diferencia es que `d3.csv()` y `d3.json()` pueden producir una matriz de objetos JSON, mientras que `d3.xml()` crea un documento XML al que es necesario acceder de un modo diferente.

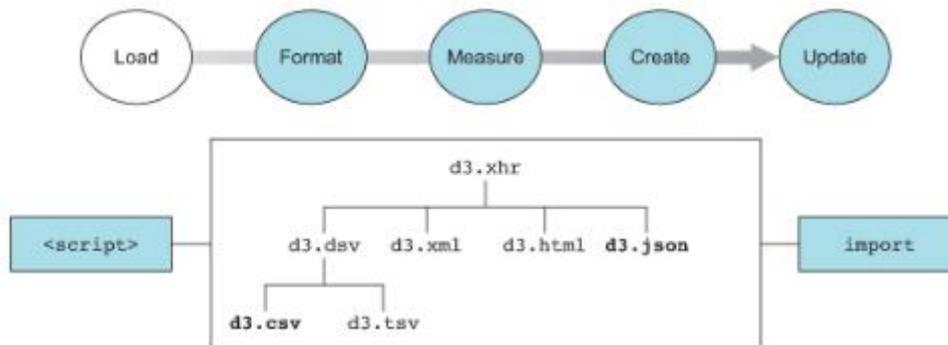


Figura 6. Carga de datos

Como se puede ver en la figura 5, el primer paso para crear una Visualización de Datos es la obtención de los datos. Para ello, se puede cargar el archivo de forma asíncronica haciendo uso de una de las diversas funciones `d3.xhr` o a través de la importación o inclusión de datos. Si los datos son correctos, cualquiera de las tres formas mencionadas es correcta, sin embargo, si se va a reemplazar la fuente de datos con una llamada API dinámica, entonces, en este caso una solicitud XHR es la mejor opción.

La figura 6 muestra las cinco funciones que se emplean para cargar los datos, que corresponden a los cinco tipos de archivos con mayor probabilidad de ser usados: `d3.text()`, `d3.xml()`, `d3.json()`, `d3.csv()` y por último, `d3.html()`.

3.2 Datos de formato

Una vez que se han cargado los conjuntos de datos, es necesario definir los métodos para que los atributos de los datos se relacionen directamente con la configuración de los elementos gráficos de color, tamaño y posición. Si bien por lo general, será necesario formatear datos cuantitativos (segundo paso en el proceso de la Visualización de Datos) para que estos puedan ser utilizados para la creación de gráficos. En la figura 7 se pueden ver diferentes colores asignados a las escalas, estos hacen referencia a la cantidad, o contenedores para anidar los datos visualmente.

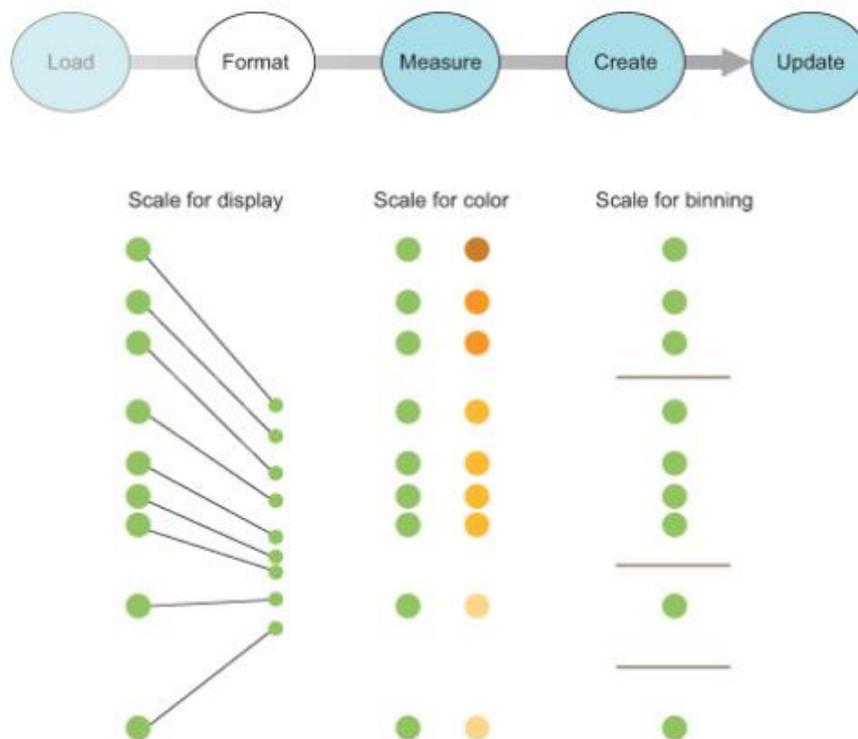


Figura 7. Formateo de los datos

A continuación, se describirán los diferentes tipos de datos que pueden ser utilizados en la Visualización de Datos:

Cuantitativo

Los datos numéricos o categóricos son los más comunes. Este tipo de datos se pueden representar de un modo efectivo con tamaño, posición o color. Un ejemplo utilizado muy a menudo puede ser las cifras de población empleados en tablas.

Categorico

Los datos categóricos son aquellos que pertenecen a grupos discretos, por lo general, representados por texto, nacionalidad o género. Es muy común, representar este tipo de datos con forma o color, pudiendo asignar las categorías a distintos colores o formas para identificar el patrón de los grupos de elementos posicionados de acuerdo con otros atributos.

Topológico

Los datos topológicos describen la relación de una pieza de datos con otra, que también puede ser otra forma de datos de ubicación. Además, los atributos topológicos se pueden representar con texto que hace referencia a valores únicos de ID o con punteros a otros objetos.

Geométrico

Los datos geométricos se asocian por lo general con los límites y las pistas de datos geográficos, como países, ríos, ciudades o carreteras. Es muy común, representar este tipo de datos con forma y tamaño, pero también se pueden transformar en otros datos, por ejemplo, en datos cuantitativos midiendo el área y el perímetro.

Temporal

Los datos temporales como las fechas y el tiempo se pueden representar empleando números durante días, meses o años, o a través de la codificación de fecha y hora específica para cálculos más complejos. El formato más común es el ISO 8601, y si sus datos se formatean como una cadena, es simple convertirlo en un tipo de fecha en JavaScript.

Crudo

Los datos crudos, libres o no estructurados, también denominados datos fuente o datos atomizados, suelen ser datos e imágenes. Este tipo de datos pueden ser transformados midiéndolos o utilizando sofisticados análisis de texto e imágenes para derivar atributos más adecuados para la Visualización de Datos.

3.3 Transformación de los datos

Al lidiar con diferentes formas de datos, será necesario realizar el cambio de formato de un tipo de dato a otro diferente, para poder representarlo mejor. La transformación de los datos puede realizarse de diferentes formas.

En este caso se verá: casting o fundición, normalización o escalado, binning o agrupamiento y anidación de datos.

Fundición o casting

Cuando se produce la carga de datos, con frecuencia se encontrarán en formato de cadena, incluso en forma de fecha, número entero, número de coma flotante o una matriz. A continuación, se visualizarán algunos ejemplos:

1. `parseInt ("77"); +"77";`
2. `parseFloat ("3.14"); +"3.14";`
3. `text = "alpha,beta,gamma"; text.split(",");`

Si se trata de entender que sucede en cada ejemplo, se encuentra:

1. Coloca la cadena 77 en el número 77 sin decimales.
2. Coloca la cadena 3.14 en el número 3.14 con decimales.
3. Divide la cadena delimitada por comas en una matriz, que no es estrictamente una operación de conversión, sino que cambia el tipo de datos.

Normalización y escalado

Los datos numéricos pocas veces se corresponden directamente con la posición y el tamaño de los elementos gráficos. Se puede emplear `d3.scale ()` para la normalización de datos en la representación en una pantalla. La primera escala que será explicada es `d3.scale().linear()`, la cual establece una relación directa entre un rango de números y otros.

Las escalas en D3.js asignan un conjunto de valores (el dominio) a otro conjunto de valores (el rango) en una relación determinada por el tipo de escala que crea.

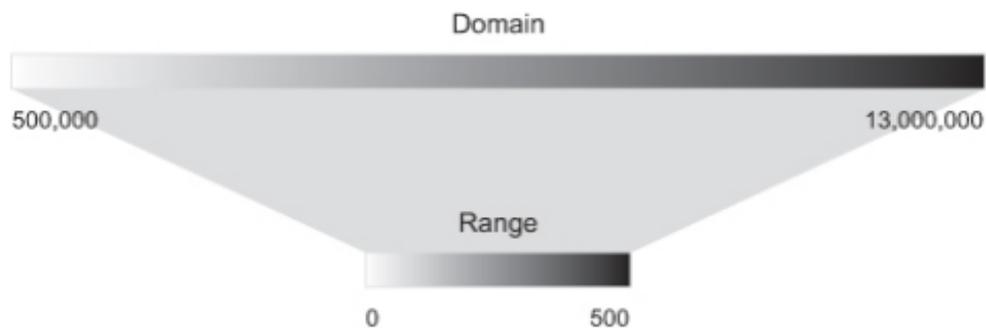


Figura 8. Escala en D3.js

A continuación, se muestran algunos ejemplos:

1. `Var newRamp =d3.scaleLinear().domain([500000,13000000]).range([0, 500]);`
2. `newRamp(1000000);`
3. `newRamp(9000000);`
4. `newRamp.invert(313);`

De nuevo se tratará de entender que sucede en cada ejemplo:

1. Se crea una rampa de modo que se instancia un nuevo objeto de escala y se establece sus valores de dominio y rango.
2. Devuelve 20, lo que permite colocar un país con población de 10.000.000 a 20px.
3. Devuelve 340.
4. La función inversión se encarga de invertir la transformación, es por ello, que en este caso se devolverá 8325000.

D3.js también permite la creación de una rampa de color, haciendo referencia a nombres de colores CSS, colores RGB () o colores hexadecimales en el campo del rango, es decir, las escalas también se pueden usar para asignar valores numéricos a bandas de color, para que sea más fácil denotar valores usando una escala de colores.

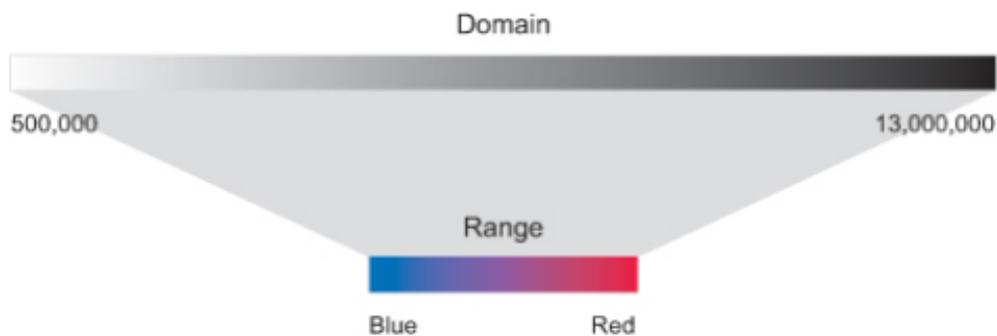


Figura 9. Escala a color en D3.js

El código para la creación de esta rampa es exactamente el mismo que el utilizado anteriormente, a diferencia de la referencia a los colores en la range matriz:

1. `Var newRamp = d3.scaleLinear(). domain([500000,13000000]). Range(["blue", "red"]);`
2. `newRamp(1000000);`
3. `newRamp(9000000);`
4. `newRamp.invert("#ad0052");`

De nuevo se tratará de entender que sucede en cada ejemplo:

2. Devuelve "#0a00f5" lo que permite realizar el dibujo de una ciudad con población de 1.000.000 como púrpura oscura.
3. Devuelve "#0d0052".
4. La función inversión solo funciona con un rango numérico, por lo que la inversión en este caso sólo devuelve NaN.

Además, se pueden emplear `d3.scaleLog()`, `d3.scalePow()`, `d3.scaleOrdinal()` para mapear un conjunto de datos. Sin olvidar, `d3.scaleTime()` que proporciona una escala lineal diseñada para tratar datos tipo fecha.

Binning o agrupamiento

La clasificación de datos cuantitativos en categorías, colocando los valores en un rango o “bin” para agruparlos es lo que permite este tipo de transformación de datos. Un método es el empleo de cuantiles, dividiendo la matriz en partes del mismo tamaño, la escala de cuantiles se llama `d3.scaleQuantile()`, y presenta la misma configuración que el resto de las escalas vistas anteriormente. Es decir, la escala Quantile toma un rango de valores y los reasigna a un conjunto de contenedores de igual tamaño.

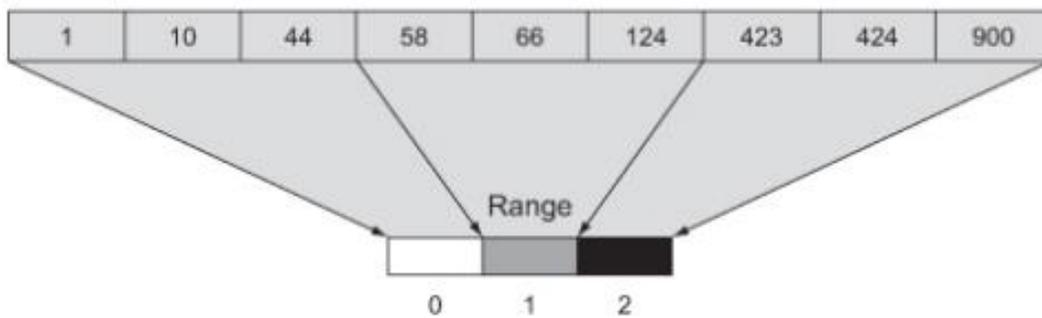


Figura 10. Binning o agrupamiento

1. `var qScaleName = d3.scaleQuantile().domain(sampleArray).range(["small", "medium", "large"]);`
2. `qScaleName(68);`
3. `qScaleName(20);`
4. `qScaleName(10000);`

De nuevo se tratará de entender que sucede en cada ejemplo:

2. Devuelve “medio”.
3. Devuelve “pequeño”.
4. Devuelve “grande”.

Anidación

Las representaciones jerárquicas de datos son útiles y no se encuentran limitadas a datos con jerarquías más tradicionales o explícitas. La función de anidación `d3.nest()`, permite que los atributos compartidos de los datos se puedan usar para clasificarlos en categorías o sub-categorías discretas.

Un ejemplo lo podemos encontrar, en el caso de que queramos agrupar tweets por parte del usuario que los creó, entonces en esta situación se emplearía la anidación:

```

2. d3.json("tweets.json", data => {
3.   var tweetData = data.tweets;
4.   var nestedTweets = d3.nest
5.     .key(d => d.user)
6.     .entries(tweetData);
7. });

```

Ahora, una vez que se han cargado los datos y se han transformado en tipos accesibles, se va a proceder a investigar los patrones de esos datos midiendo los datos.

3.4 Medición de datos

Una vez que se ha cargado la matriz de datos, el primer paso a realizar es la medición y ordenación de dicha matriz. Es muy importante conocer la distribución de valores de atributos particulares, así como los valores mínimos, máximos y los nombres de los atributos.

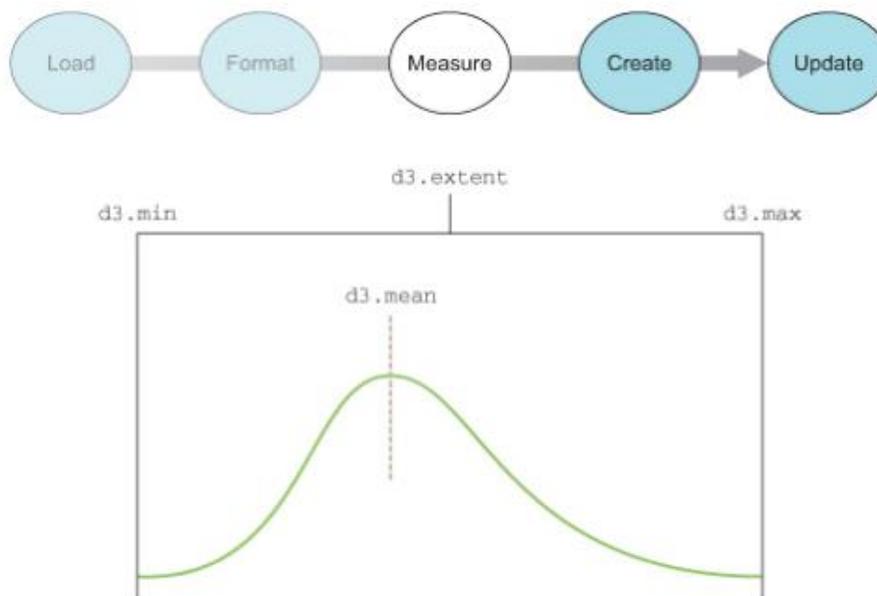


Figura 11. Medición de datos

Una vez que se han formateado los datos, será necesario que los datos sean medidos para asegurarnos que los gráficos que se creen estén dimensionados y posicionados adecuadamente, en función de los parámetros del conjunto de datos. Para ello se empleará todo el tiempo `d3.extent()`, `d3.min()`, `d3.mean()` y `d3.max()`.

D3.js proporciona un conjunto de funciones de matriz de forma que facilita la comprensión de los datos. Casi todas las funciones de medición de D3.js siguen el mismo patrón. En primer lugar, se debe asignar la matriz y una función de acceso para el valor que desea medir. Para entenderlo mejor, se tomará una matriz de números:

```
var testArray = [88,10000,1,75,12,35];
```

Una vez que se ha visto la matriz de números de la que se parte, el descriptor de acceso necesita apuntar al elemento en sí:

1. `d3.min(testArray, el => el);`
2. `d3.max(testArray, el => el);`
3. `d3.mean(testArray, el => el);`

Se tratará de entender que sucede en cada ejemplo:

1. Devuelve el valor mínimo de la matriz, en este caso 1.
2. Devuelve el valor máximo de la matriz, en este caso 10000.
3. Devuelve el valor medio de la matriz, en este caso 1701,833333333335.

También se puede medir datos no numéricos como texto empleando la función `d3.length()`, la cual se emplea para cadenas y matrices.

Una vez que se han cargado, formateado y medido todos los datos, el siguiente paso es la Visualización de Datos. Por lo que es necesario emplear las selecciones y las funciones que van los propios datos.

3.5 Creación de gráficos

En la creación de gráficos en D3.js, se emplean selecciones que vinculan los datos a elementos DOM.



Figura 12. Creación de gráficos

Las selecciones se utilizan para realizar cambios en la estructura y el aspecto de la página web que se está realizando con D3.js. La selección consiste en uno o más elementos en el DOM, así como también los datos, si los hay, asociados a ellos. También es posible realizar la creación o eliminación de elementos, haciendo uso de selecciones y cambiar el estilo y el contenido [7].

3.6 Estilo de presentación de datos, atributos y contenido

Por último, se procederá a la Visualización de Datos reales, para ello existe un grandísimo número de métodos diferentes de presentación que se irán explicando uno a uno, aquellos considerados como los más importantes.

Polar Clock

Es un inusual reloj basado en coordenadas polares, que indica al usuario la fecha, indicando mes, día, día de la semana, horas, minutos y segundos.



Figura 13. Polar Clock

Floating Landmasses

Consiste en la recreación de dos esferas concéntricas para dar la apariencia de masas terrestres flotantes. Se presenta en continuo movimiento de rotación, es decir, de la Tierra sobre sí misma y permite cambiar la escala y el ángulo del clip de una proyección ortográfica.



Figura 14. Floating Landmasses

Koalas to the max

Inicialmente se parte de una circunferencia, de modo que cuando que se pasa el puntero por ella se divide en otras 4 circunferencias, y seguidamente, cada vez que se pasa de nuevo el puntero por cada circunferencia creada, se generan otras cuatro nuevas circunferencias. La evolución se puede ver a continuación:

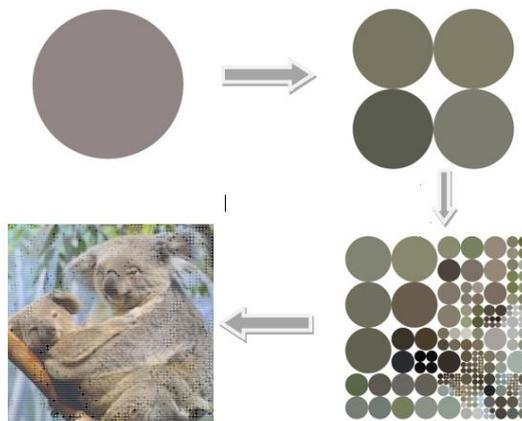


Figura 15. Koalas to the max

Blox Plots

Los diagramas de caja y bigotes, que es como son los llamados box plots, emplean glifos que resumen una distribución cuantitativa con cinco estadísticas estándar: el valor más pequeño, la mediana, el cuartil inferior, el cuartil superior y el valor más grande, de forma que el usuario puede reconocer fácilmente las diferencias entre distribuciones.

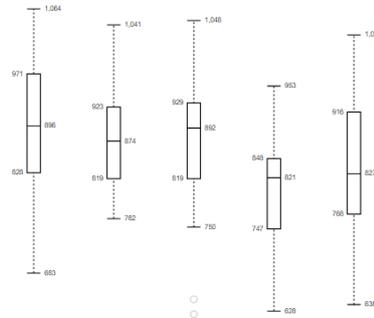


Figura 16. Box Plots

Bubble Charts

Los gráficos de burbujas codifican datos en el área de círculos. Aunque son menos perceptualmente precisos que los gráficos de barras pueden empaquetar cientos de valores en un espacio pequeño. Una variación de este tipo de grafico es Simple Bubble Chart o Interactive Bubble Chart.

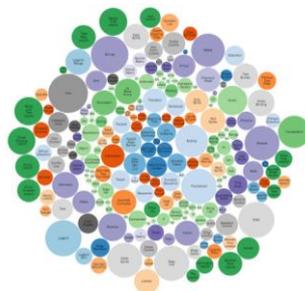


Figura 17. Bubble Charts

Bullet Charts

Los gráficos bala o gráficos de viñeta representan una variación en un gráfico de barras, de forma que comparan una medida cuantitativa dada como por ejemplo ingresos, con rangos cualitativos como por ejemplo satisfactorio, bueno, medio o malo. Una variación de este tipo de gráficos es Fitbit Bullet Chart.

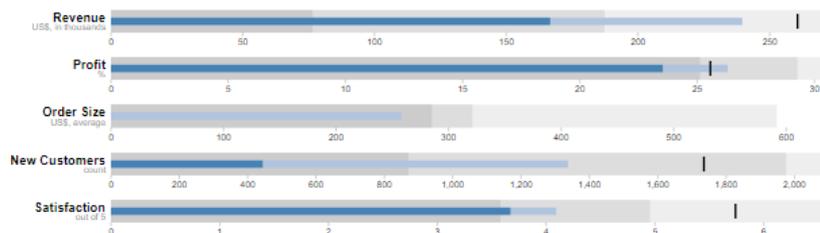


Figura 18. Bullet Charts

Calendar View

En este tipo de gráficos los valores se visualizan como celdas de colores por día. Los días se organizan en columnas por semana, y luego se agrupan por mes y año. Una variación de este tipo de gráficos es Alternative Calendar View.

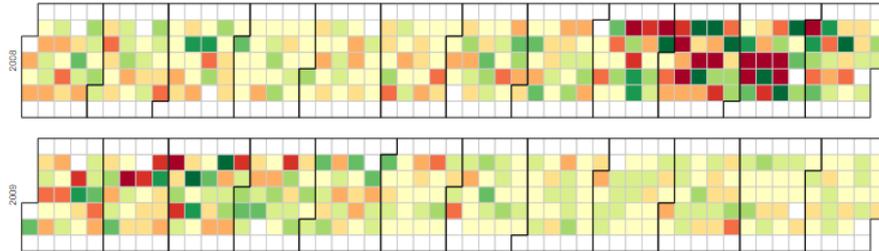


Figura 19. Calendar View

Non-contiguous Cartogram

A diferencia de los mapas de coropletas, los cartogramas codifican los datos usando el área del color, lo que da como resultado límites geográficos distorsionados. En el ejemplo, los estados se vuelven a escalar alrededor de su centroide, preservando la forma local pero no la topología. Una variación de este tipo de mapa es el Zoomable Map o Hip Replacement by State.

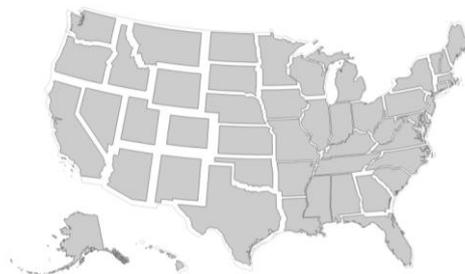


Figura 20. Non-contiguous Cartogram

Chord Diagram

Chord Diagram es gráfico sencillo, dado que los diagramas de acordes muestran las relaciones dirigidas entre un grupo de entidades. Algunas de las variaciones de este tipo de gráficos son el Euro Debt Crisis, Global Refugee Flows- 2 Dynamic Chord Diagrams, Chord Viz.



Figura 21. Chord Diagram

Dendrogram

Un dendrograma es un diagrama de enlace de nodo que coloca los nodos de hoja de árbol a la misma profundidad. En el ejemplo las clases (nodos hojas) se encuentran alineadas en el borde derecho, con los paquetes (nodos internos) a la izquierda.

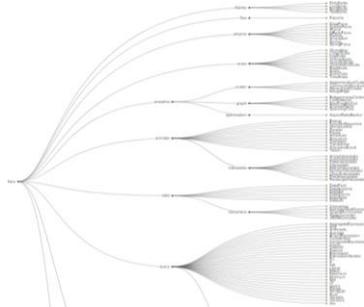


Figura 22. Dendrograma

Force-Directed Graph

Los gráficos orientados a la fuerza es una simulación física de partículas cargadas y resortes que enlaza aquellos elementos de un determinado tema que se encuentran relacionados más cerca, mientras que aquellos elementos que se encuentran menos relacionados estarán situados más alejados. Algunas de las variaciones de este tipo de gráficos son Interactive Force Directed Graph, Force Directed Graph with Search and Subset View o Ontology Visualization.

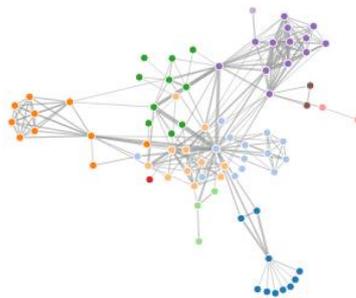


Figura 23. Force-Directed Graph

Circle Packing

Los diagramas de alojamiento usan contención para representar la jerarquía. A pesar de que el empaquetado en círculo no es tan eficiente desde el punto de vista del espacio como un mapa de árbol, es mejor que revele la jerarquía.

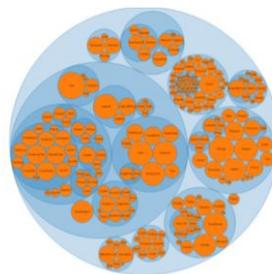


Figura 24. Circle Packing

Stacked Bars

Este tipo gráfico permite la visualización en dos modos, el primer modo “agrupado”, y el segundo modo “apilado”.

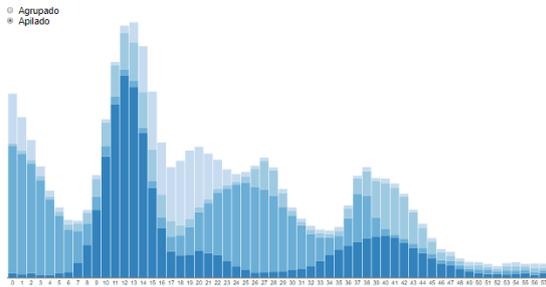


Figura 26. Stacked Bars Apilado

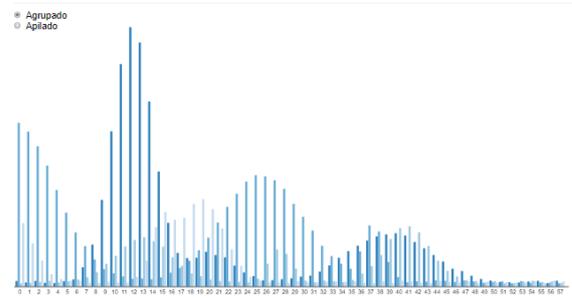


Figura 25. Stacked Bars Agrupado

Streamgraph

Para datos continuos tales como series de tiempo, se puede emplear un diagrama de flujo en lugar de barras apiladas. El ejemplo muestra transacciones de ruta para interpolar entre diferentes diseños.

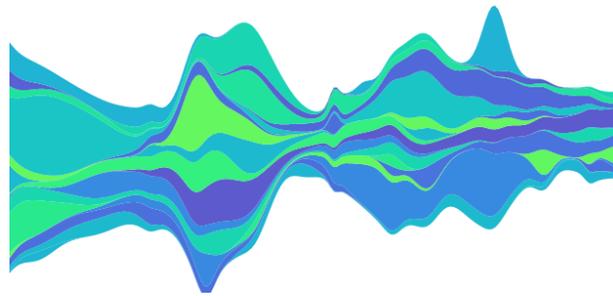


Figura 27. Streamgraph

Sunburst o Bilevel Partition

Un resplandor solar es similar al Treemap, excepto por el hecho de que emplea diseño radial. El nodo raíz del árbol está en el centro, con hojas en la circunferencia. El área (o ángulo, dependiendo de la implementación) de cada arco corresponde a su valor. Algunas de las variaciones de este tipo de gráficos son Hierarchie Sunburst, Bilevel Partition with Tooltips o Zoomable Unilevel Partition.



Figura 28. Sunburst o Bilevel Partition

Node-Link Tree

El diseño d3.tree implementa el algoritmo Reingold-Tilford para una disposición ordenada y eficiente de los nodos en capas. La profundidad de los nodos se calcula por la distancia desde la raíz, dando lugar a una apariencia irregular. Las orientaciones cartesianas también son compatibles. Una variación de este tipo de gráfico es Node Focusable Tree.

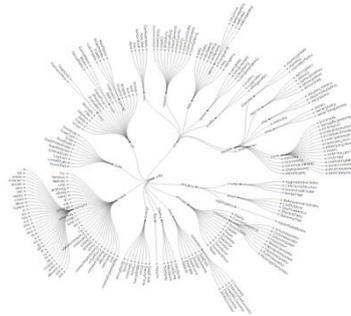


Figura 29. Node-Link Tree

Treemap

Un Treemap subdivide recursivamente el área en rectángulos; el área de cualquier nodo en el árbol es proporcional a su valor. Además, este tipo de gráfico permite la visualización en dos modos, el primer modo “Tamaño”, y el segundo modo “Contar”. Este tipo de gráfico presenta tres variaciones: Zoomable Treemap, Zoomable Partition Layout y Sequences Sunburst.



Figura 30. Treemap tamaño



Figura 31. Treemap contar

Voronoi Tessellation

Este tipo de gráfico presenta forma de constelación, y se crea al unir los puntos entre sí, trazando las mediatrices de los segmentos de unión. Las intersecciones de estas mediatrices determinan una serie de polígonos en un espacio bidimensional.

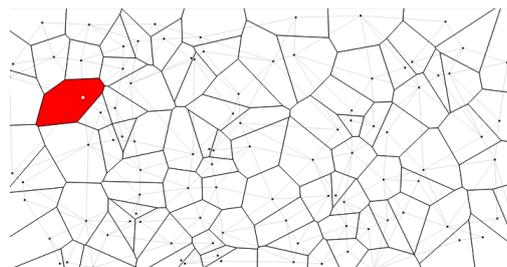


Figura 32. Voronoi Tessellation

Hierarchical Edge Bundling

Es una representación radial, de forma que el usuario puede adaptar la tensión de los nodos de enlace. Una variación de este tipo de diagramas son Language Network o Dependency.

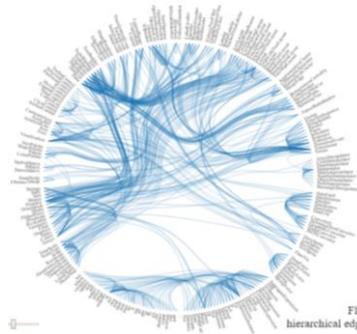


Figura 33. Hierarchical Edge Bundling

Voronoi Diagram

Los diagramas de Voronoi es un método sencillo de interpolación, basados en la distancia euclidiana, especialmente apropiada cuando los datos son cualitativos. Se crean al unir los puntos entre sí, trazando las mediatrices de los segmentos de unión.

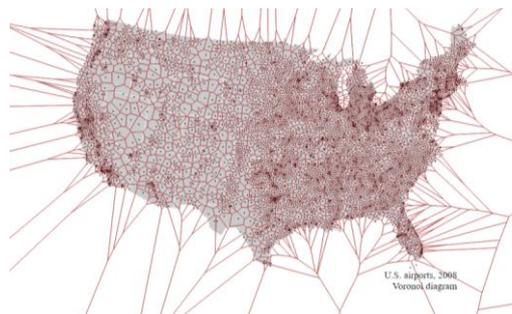


Figura 34. Voronoi Diagram

Symbol Map

Este tipo de gráficos permite visualizar en función de territorios, como pueden ser los diferentes estados que forma el país de U.S.A, el tema que el usuario desee, en el ejemplo se puede ver como en el año 1999, en función del número de empleos creados o eliminados, se crean nodos de mayor o menos tamaño.

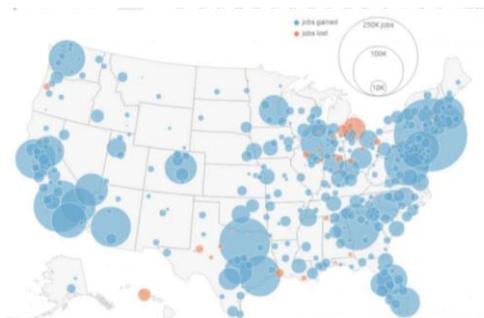


Figura 35. Symbol Map

Parallel Coordinates

Las coordenadas paralelas permiten mostrar un conjunto de puntos en un espacio n-dimensional, se dibuja un fondo que consta de n líneas paralelas, por lo general verticales e igualmente espaciadas. Es una buena opción para comparar muchas variables juntas y ver las relaciones entre ellas.

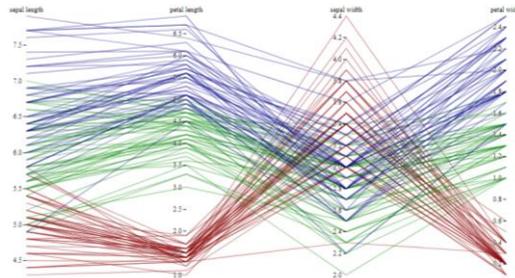


Figura 36. Parallel Coordinates

Scatterplot Matrix

Las matrices de diagramas de dispersión es un buen método para determinar si existe una correlación lineal entre múltiples variables. Resulta útil para identificar las variables específicas que pueden tener correlaciones similares a sus datos genómicos o proteómicos. Algunas de las variaciones de este tipo de gráficos son Multivariate Data Exploration with Scatterplots o Scatterplot as Barchart.

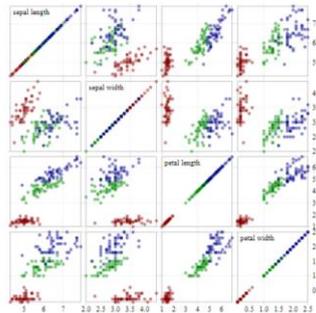


Figura 37. Scatterplot Matrix

Hierarchical Bars

Las barras jerárquicas es un gráfico de barras horizontales en capas donde cada vez que se pulsa sobre una de las barras permite profundizar y extraer los componentes de cada elemento.

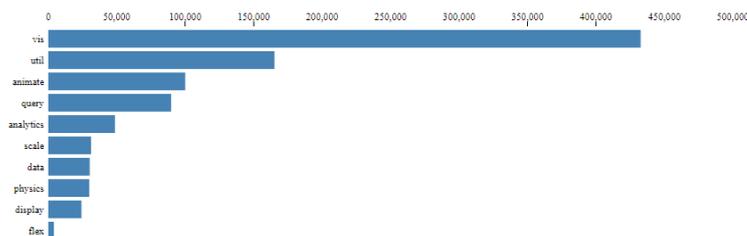


Figura 38. Hierarchical Bars

Epicyclical Gears

Engranaje epicicloidal o planetario es un sistema de engranaje que consiste en uno o más engranajes externos o engranajes planetarios, que giran alrededor de un engranaje central o solar. Este tipo de sistema incorpora también el uso de un engranaje anular externo o anillo, que se acopla con los engranajes planetarios, de forma que el usuario puede elegir el marco de referencia a sol, planetas o anular.

Anular: el engranaje externo permanece inmóvil, mientras que los internos rotan, tanto el central de tamaño inferior como los tres periféricos.

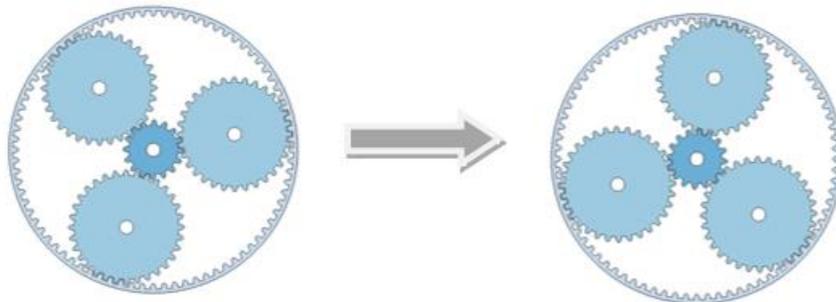


Figura 39. Epicyclical Gears anular

Solar: el engranaje central de tamaño inferior permanece inmóvil, mientras que los tres engranajes internos rotan, así como el exterior, y todos en la misma dirección.

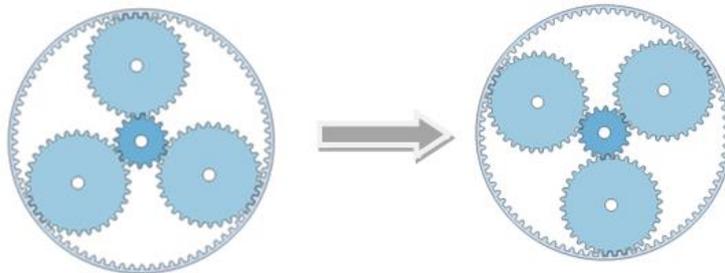


Figura 40. Epicyclical Gears solar

Planetas: Todos los engranajes giran, tanto el exterior, como los internos, que incluye el central de menor tamaño y los otros tres periféricos, sin embargo, en este caso únicamente rotan, no se desplazan.

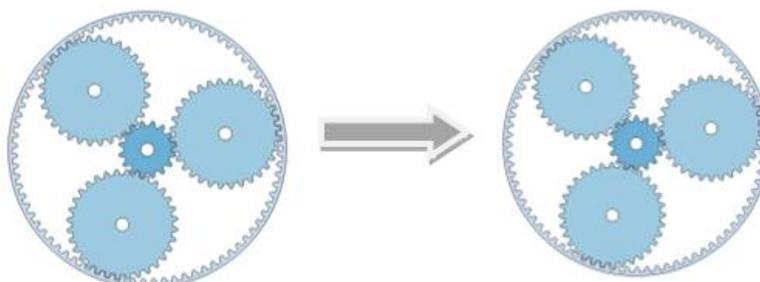


Figura 41. Epicyclical Gears planetas

Collision Detection

Este tipo de gráfico permite repeler los nodos cada vez que se acerca el puntero al conjunto, es decir, los nodos tratan de evitar el puntero y al mismo tiempo unirse junto al resto de nodos.

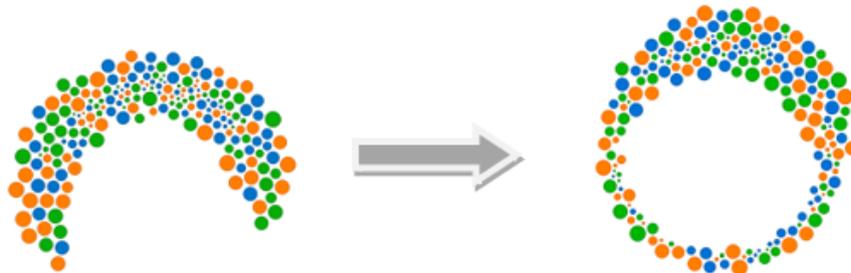


Figura 42. Collision Detection

Collapsible Force Layout

En este tipo de diagrama los nodos internos son colapsables, donde los nodos de las hojas se muestran en naranja, a diferencia de los nodos internos que se muestran en azul. Al coger un nodo interno (sin arrastrar), ese nodo se expande o contrae, alternando la visibilidad de los nodos descendientes. Algunas de las variaciones de este tipo de gráficos son el Force-based Label Placement, Force Directed Tag Explorer, 3D Force Layout, Force-directed Graph with Elliptic Forces, OrgoShmorgo.

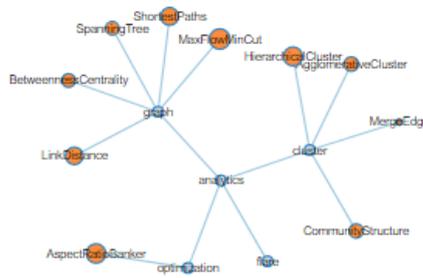


Figura 43. Collapsible Force Layout

Spacetime

Es una representación de la época del año empleando el Sol, la Tierra y la Luna. La posición de la Tierra en órbita alrededor del Sol indica la hora del año. La Tierra en si es un gráfico circular que indica el segundo del día, y la Luna en órbita alrededor de la Tierra indica la hora del mes.

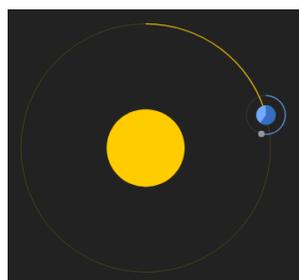


Figura 44. Spacetime

Azimuthal Projections

La proyección acimutal es la que se consigue proyectando una porción de la Tierra sobre un plano tangente a la esfera en un punto seleccionado, obteniéndose la visión que se lograría ya sea desde el centro de la Tierra o desde un punto del espacio exterior. Existen 5 tipos de proyecciones acimutales que se verán a continuación:



Figura 45. Azimuthal Projections equidistante



Figura 46. Azimuthal Projections equalarea



Figura 47. Azimuthal Projections gnomónico



Figura 49. Azimuthal Projections ortográfico



Figura 48. Azimuthal Projections esterográfica

Choropleth

Los mapas de coropletas muestran áreas geográficas divididas o regiones que están coloreadas, sombreadas o modeladas en relación a una variable de datos. Esto proporciona una forma de visualizar valores en un área geográfica, que puede mostrar variaciones o patrones en una ubicación. En este caso muestra la tasa de desempleo en una región. Una variación de este tipo de mapa es Choropleth with SVG Filter.

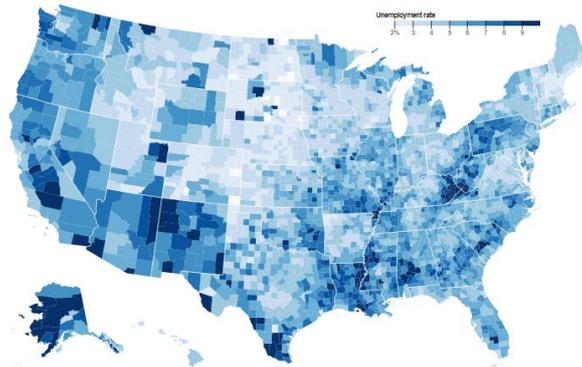


Figura 50. Choropleth

Drag and Drop Collapsible Tree Layout

Este tipo de diagrama reúne varios tipos de gráficos de trabajo con árboles. La funcionalidad de la panoramización puede realizarse arrastrando una parte vacía del SVG o arrastrando un nodo hacia un borde. El zoom se realiza haciendo doble clic en una parte vacía del SVG o desplazando la rueda del ratón. La ventaja de este diagrama es que calcula automáticamente sus tamaños tanto horizontal como vertical, por lo que puede adaptarse entre muchos nodos que están presentes en la vista a muy pocos. Este tipo de diagrama presenta múltiples variaciones entre las que se encuentra: Collapsible Tree Layout, Dendrogram (visto anteriormente), Federal Budget, Collapsible 2-Way Tree Layout o Bi-directional Drag and Zoom Tree on D3 development (es igual, pero en orientación horizontal).

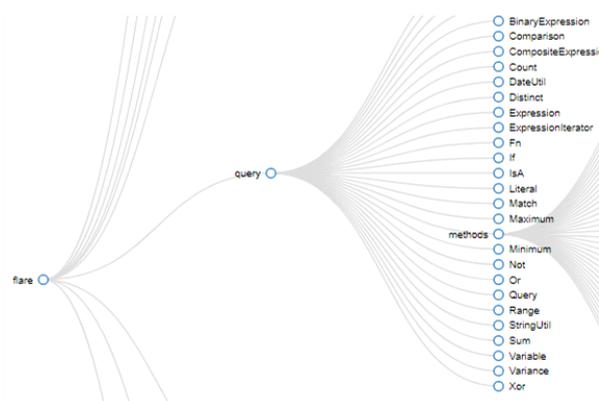


Figura 51. Drag and Drop Collapsible Tree Layout

Rotating Cluster Layout

Los diagramas de enlace de nodo colocan los nodos de hoja de árbol a la misma profundidad, pero ya expansionados. La característica principal es la forma circular que presentan y que se puede girar el gráfico 360°. Este diagrama presente una variación: US Trade Deficit.

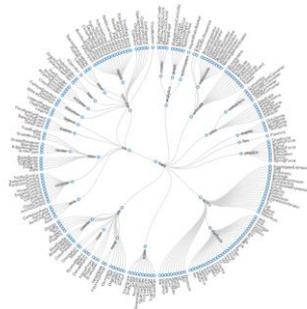


Figura 52. Rotating Cluster Layout

Sankey Diagram

Los diagramas de Sankey visualizan la magnitud del flujo entre nodos en una red. El ejemplo muestra la producción y consumo de energía de un país, donde los suministros de energía se encuentran a la izquierda y las demandas a la derecha. Los nodos intermedios muestran como la energía se convierte y se transmite antes de que se consuma. Una variación de este tipo de diagrama es Bi-directional Hierarchical Sankey, Vertical Sankey o Re-usable Sankey.

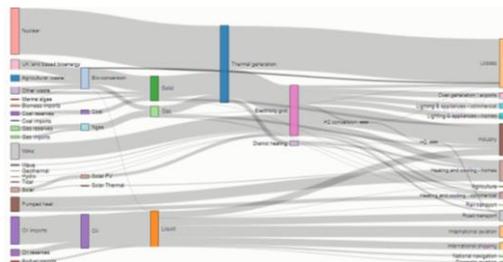


Figura 53. Sankey Diagram

Confusion Matrix

La matriz de confusión permite la visualización del desempeño de un algoritmo que se emplea en aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real.

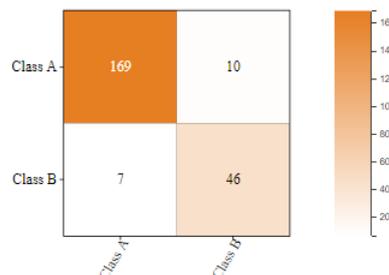


Figura 54. Confusion Matrix

Co-occurrence Matrix

Una red puede ser representada por una matriz de adyacencia, donde cada celda ij representa un borde desde el vértice i hasta el vértice j . La efectividad de un diagrama matricial depende en gran medida del orden de filas y columnas. Este tipo de diagramas puede ser visualizado en 3 modos diferentes: por nombre, frecuencia o cluster.

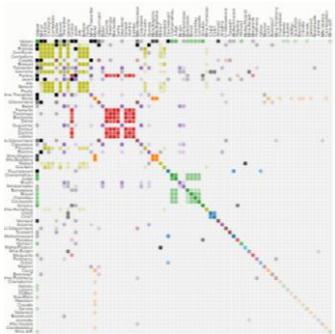


Figura 57. Co-ocurrence Matrix nombre

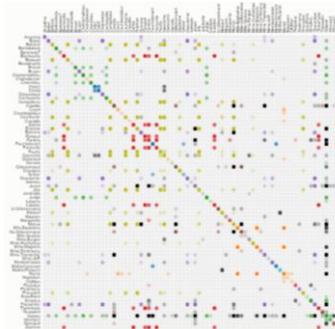


Figura 56. Co-ocurrence Matrix frecuencia

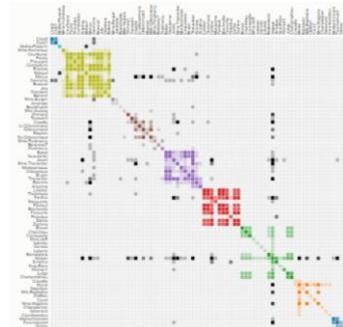


Figura 55. Co-ocurrence Matrix cluster

Animated Bézier Curves

Las animaciones muestran cómo se construye una curva paramétrica de Bézier. El parámetro t varía de 0 a 1. En el caso más simple, una curva de Bézier de primer orden, la curva es una línea recta entre los puntos de control. Si es de segundo orden (cuadrática), primero se tiene dos puntos intermedios que son t a lo largo de las líneas entre los tres puntos de control. Si se realiza el mismo paso de interpolación de nuevo, se encuentra otro punto que es t a lo largo de la línea entre los dos puntos intermedios. Al trazar este último punto se obtiene una curva cuadrática de Bézier. Los mismos pasos pueden repetirse para órdenes superiores.

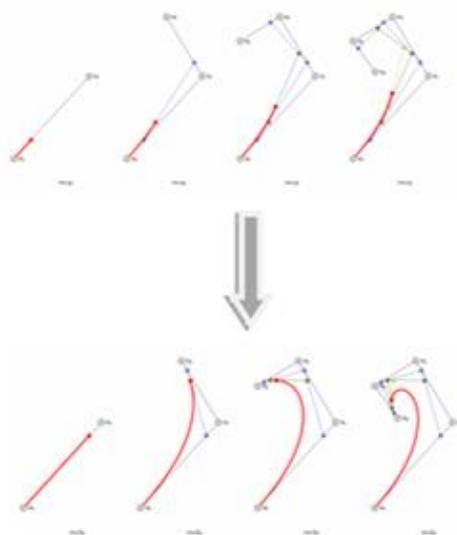


Figura 58. Animated Bézier Curves

Zoomable Sunburst

Un árbol de rayos solares es una visualización radial llena de espacio, donde los colores empleados para los nodos internos se calculan automáticamente tomando el promedio de los dos nodos secundarios en el espacio de color HSL. Además, si una etiqueta tiene más de una palabra, se divide en dos líneas y se coloca equidistante al centro del segmento. Una variación de este tipo de gráfico es Sunburst for your skill map.



Figura 59. Zoomable Sunburst

Collatz Graph

Este tipo de gráfico sigue la conjetura de Collatz que se puede enunciar como:

1. Toma cualquier número natural, n .
2. Si n es par, se divide por 2.
3. De lo contrario, si n es impar. Se multiplica por 3 y se suma 1.
4. Se repite de forma indefinida.

La conjetura es que siempre se llega a 1, sin importar con qué número comiences. En este punto, por supuesto, se termina en un ciclo interminable de 1 a 4, a 2 y de nuevo a 1.

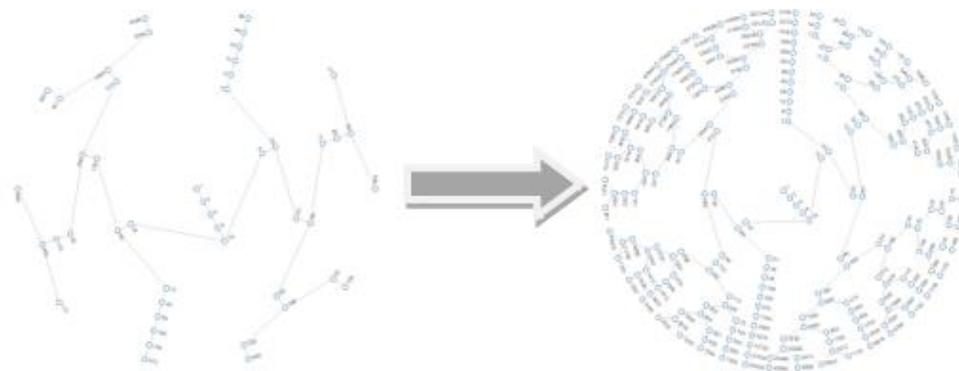


Figura 60. Collatz Graph

Parallel Sets

En este tipo de diagramas para cada dimensión se muestra una barra horizontal para cada una de sus posibles categorías. El ancho de la barra indica el número absoluto de coincidencias para esa categoría. Comenzando por la primera dimensión, cada una de sus categorías está conectada a una serie de categorías en la siguiente dimensión, que muestra cómo se subdivide esa categoría. Esta subdivisión se repite recursivamente produciendo un árbol de cintas.

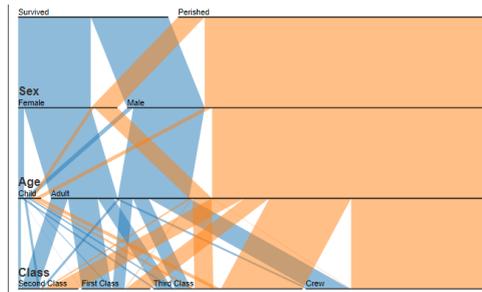


Figura 61. Parallel Sets

World Cloud Generator

El algoritmo de diseño para el posicionamiento de palabras sin superposición se ejecuta de forma asíncrona, con un tamaño de paso de tiempo configurable, lo que hace posible la animación de palabras. Este algoritmo es muy sencillo pues en la colocación de las palabras, comienza por la más importante.



Figura 62. World Cloud Generator

D3 Mobile Application

Esta aplicación de móvil basada en web que incorpora varios gráficos y componentes vizuly optimizados para dispositivos móviles en una aplicación móvil altamente interactiva que se ejecuta en el navegador. Además, permite la Visualización de Datos basada en la web, o mezclar y combinar los diversos componentes en otros proyectos.



Figura 63. D3 Mobile Application

Halo

Esta visualización única permite mostrar conexiones bilaterales a través de una red de nodos empleando un diseño de grupo de paquetes junto con un diseño de arco de cuerda.

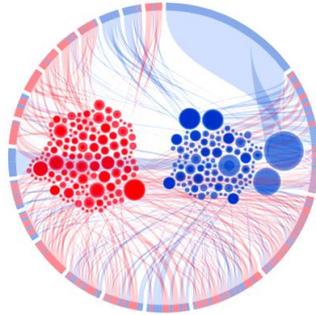


Figura 64. Halo

Corona Chart

Este gráfico radial admite gráficos radiales angulares y curvos y viene en varias opciones de diseño diferentes. También viene incluido con varias máscaras distintivas.

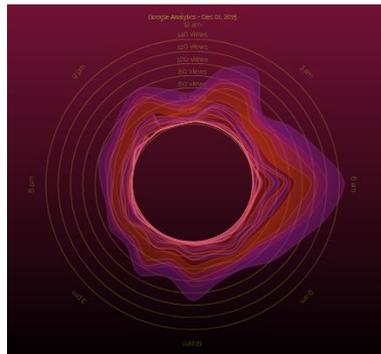


Figura 65. Corona Chart

Rounded Rectangles

Este tipo de gráficos muestra rectángulos redondeados animados. Los datos asociados con cada rectángulo son su posición central y ángulo de rotación en grados. En cada tic de la animación, los rectángulos se desplazan hacia la ubicación del ratón y giran libremente.

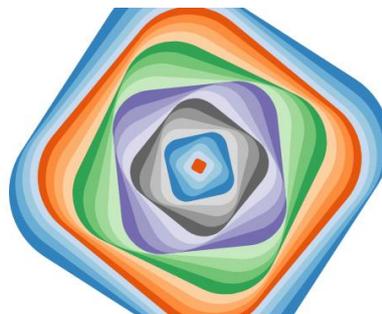


Figura 66. Rounded Rectangles

Cubism.js

Cubism.js es un plugin D3 para visualizar series de tiempo. El cubismo obtiene los datos de la serie de tiempo de forma incremental: después de la visualización inicial, el cubismo reduce la carga del servidor al sondear solo los valores más recientes.

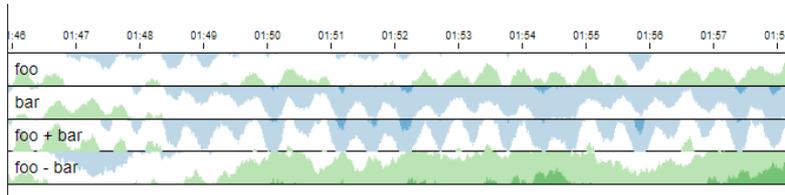


Figura 67. Cubism.js

Binary Tree with transitions

Un árbol implementado usando D3, estructuralmente es un árbol binario y cada rama se genera algorítmicamente. Se incorpora algo de aleatoriedad para que se vea más como un árbol real. Esta pieza difumina la distinción entre arte generativo y Visualización de Datos, aunque no emplea datos externos, el código genera datos para cada rama y vincula estos datos a los elementos de la línea SVG usando D3.js.

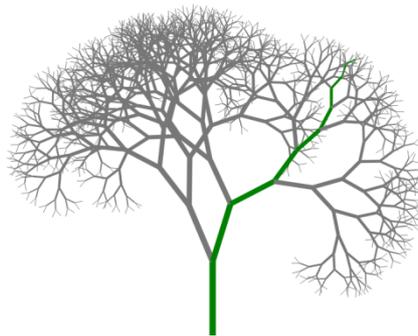


Figura 68. Binary Tree with Transitions

Site Concept Browser

Este tipo de diagramas permiten explorar sitios por tema relacionado en lugar de hacerlo jerárquicamente. Además, permite mezclar posicionamiento fijo y dinámico, de manera que gracias a esta forma se puede crear una lista de páginas que permanecen donde están, mientras que los nodos de temas flotan según sea necesario.

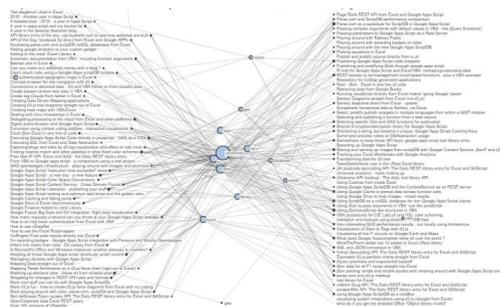


Figura 69. Site Concept Browser

Google Calendar Like Visualization

Es un simple calendario de Google como pantalla para visualizar datos temporales a través de los distintos meses. Se generan datos aleatorios para cada día y se muestran en la celda apropiada de la grilla. Los días que caen en mes anterior o posterior están sombreados en un color diferente de los días que caen en el mes actual.

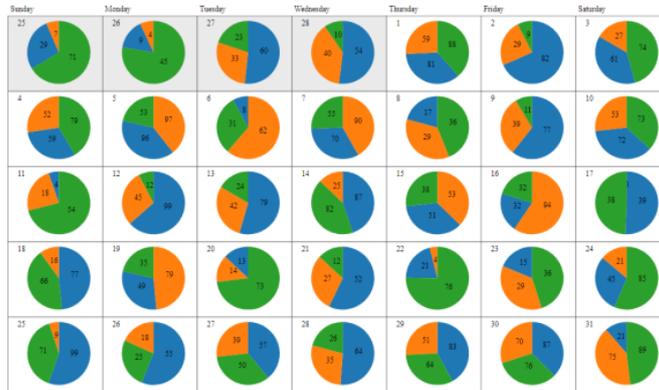


Figura 70. Google Calendar Like Visualization

Bar Chart with Tooltips

Un gráfico de barras presenta datos categóricos con barras rectangulares con alturas o longitudes proporcionales a los valores que representan. Las barras se pueden trazar vertical u horizontalmente. Algunas de las variaciones de este tipo de gráficos son Dual-Scale Bar Chart, Condegram Spiral Plot (que es un gráfico de barras pero con forma radial), 3D Honeycomb Bar Chart, 3D Bar Chart, Stacked Bar Chart, Normalized Stacked Bar Chart, Grouped Bar Chart, Bar Chart with Negative Values, Stacked Bar Chart on Time Scale, 100% Stacked Bars, Diverging Stacked Bar Chart o Multiseries Floating Bar Chart.

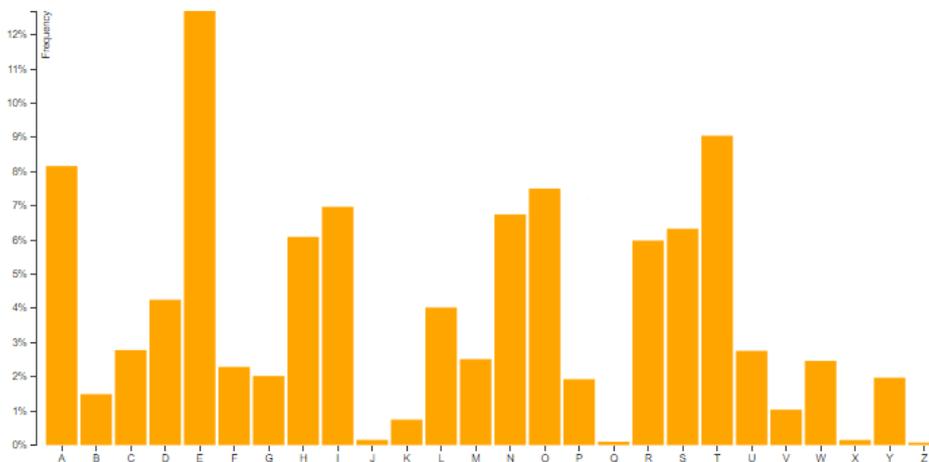


Figura 71. Bar Chart with Tooltips

Compare Time Series with Irregular Interval

Este tipo de gráficos muestra cómo se puede hacer rodar un componente de gráficos con d3.js para necesidades muy específicas. El requisito fundamental es comparar valores analógicos continuos de series de tiempo junto con algunas entradas digitales que provienen de diferentes canales con intervalos de tiempo irregulares.

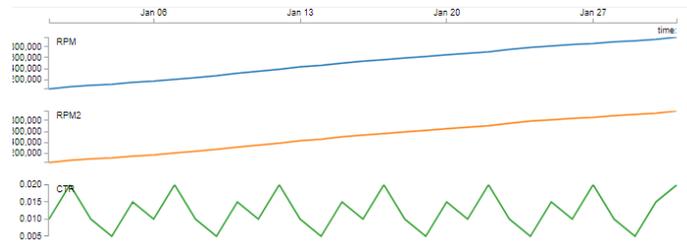


Figura 72. Compare Time Series with Irregular Interval

Heat map

Un mapa de calor es una representación gráfica de los datos, donde los valores individuales contenidos en una matriz se representan como colores. Además, los datos pueden aparecer representados en modo: cluster, por nombre de sonda y contraste de nombre. Una variación de este tipo de mapa es Days-Hours Heatmap, SOM hexagonal Heatmap o Community Pop-Culture References Heatmap.



Figura 74. Heat Map cluster



Figura 73. Heat Map nombre de sonda y contraste de nombre

Density and Quantile Graphs

El gráfico que se ve a la izquierda es la función de densidad, mientras que el de la derecha se obtiene al dividir la función de densidad horizontalmente para que todas las piezas tengan el mismo área y luego cambiar el tamaño para tener el mismo ancho y por tanto la misma altura.

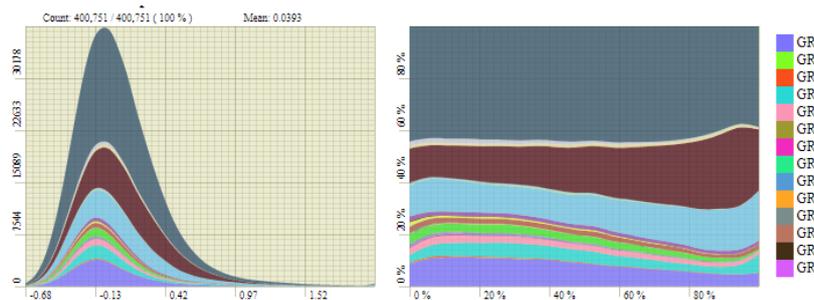


Figura 75. Density and Quantile Graphs

Aster Plot

El gráfico de Aster muestra sectores de sectores como longitudes que se extienden hacia el borde (0 en el interior a 100 en el exterior). Los anchos de los sectores de tarta representan el peso de cada pie, que se utiliza para llegar a una media ponderada de las puntuaciones de longitud en el centro. Una variación de este tipo de gráfico es Spinning Pie Chart.

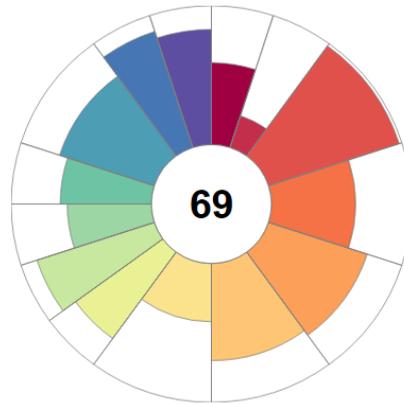


Figura 76. Aster Plot

Smallest Convex Polygon

Se trata de un algoritmo para encontrar el polígono convexo más pequeño que contiene los puntos dados. De forma que el algoritmo funciona de la siguiente manera:

1. Se encuentra el punto más alto A.
2. Se busca un punto B de manera que AB haga el ángulo más pequeño con la dirección x positiva.
3. Se busca un punto Z tal que YZ hace el ángulo más pequeño con XY donde X e Y son los dos últimos puntos encontrados (Y después de X). Se debe incluir el punto inicial A en la búsqueda. Cuando la búsqueda termina en A, entonces la ruta está completa.

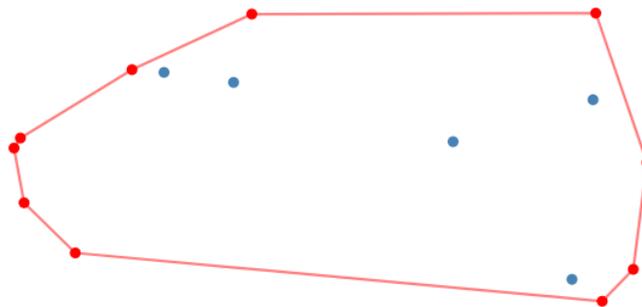


Figura 77. Smallest Convex Polygon

Trend Chart

El gráfico tendencias muestra los tiempos de carga entre lanzamientos de una aplicación web hipotética. Es una mezcla entre un gráfico de líneas y un gráfico de aéreas, con el tiempo de carga mediano diario indicado por la línea y los rangos de percentiles indicados por aéreas circundantes.

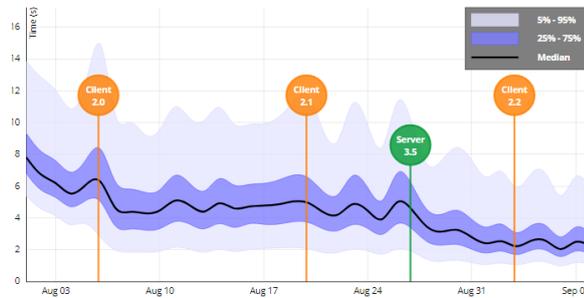


Figura 78. Trend Chart

Radial Boxplot

Los diagramas de caja son una forma excelente de resumir rápidamente las distribuciones de probabilidad usando una alta proporción de datos. En este caso, se juega con la idea de un diagrama de caja para datos continuos, en particular, datos cíclicos.

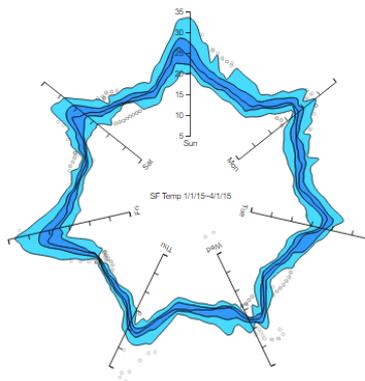


Figura 79. Radial Boxplot

Visualization of a Genetic Algorithm

Los algoritmos genéticos son uno de los métodos de optimización estándar. Funcionan mediante la evolución de una población de soluciones de prueba hacia mejores soluciones empleando operadores genéticos de inspiración biológica.

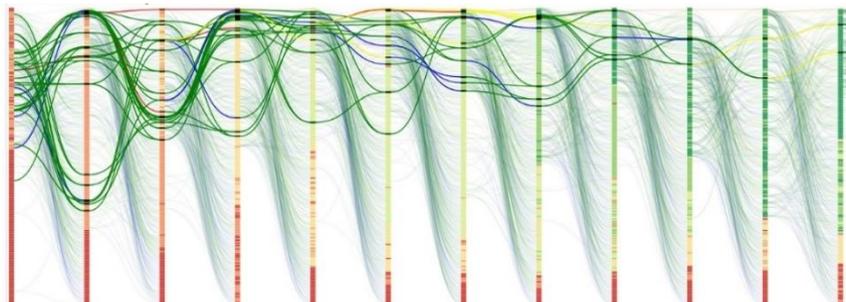


Figura 80. Visualization of a Genetic Algorithm

D3 Rectangular Area Chart

Una tabla de área rectangular presenta el valor máximo configurable, de manera que el gráfico emplea el valor máximo de los datos que se proporcionan o un valor máximo que el usuario especifica. Esto es especialmente útil si se están utilizando varios gráficos al mismo tiempo y se desea que sus tamaños relativos sean los mismos independientemente de los datos.

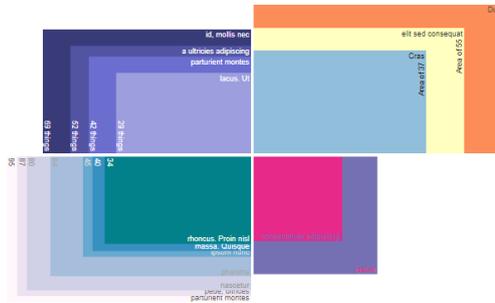


Figura 81. D3 Rectangular Area Chart

Cluster Purity Visualizer

En este tipo de gráficos cada círculo representa un grupo, cada grupo tendrá n-número de elementos de clase. Si un clúster tiene solo elementos de clase única, entonces un clúster puro. Se puede analizar en función de la longitud o del ángulo variable. Por defecto los clústeres se ordenan en orden decreciente.

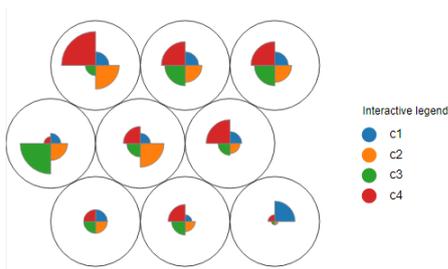


Figura 83. Cluster Purity Visualizer longitud

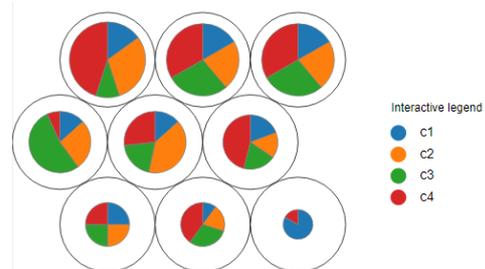


Figura 82. Cluster Purity Visualizer ángulo

Area Chart

Un gráfico de área muestra datos gráficamente cuantitativos. Estos gráficos se basan en un gráfico de líneas, de forma que se conectan un conjunto de puntos a través de una línea, con toda el área llena debajo de la línea. Una variación de este tipo de grafico es Bivariate Area Chart o Stacked Area Chart.

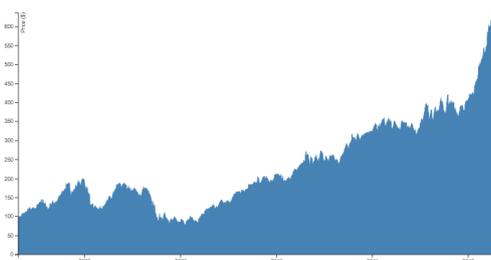


Figura 84. Area Chart

Line Chart

Un gráfico de líneas muestra información como una serie de puntos de datos llamados “marcadores” conectados a través de segmentos de línea recta. Una variación de este tipo de gráfico es Multi-Series Line Chart.



Figura 85. Line Chart

Scatterplot

Un diagrama de dispersión es un diagrama matemático que emplea coordenadas cartesianas para mostrar los valores de típicamente dos variables para un conjunto de datos. Los datos son mostrados como una colección de puntos, cada uno con el valor de una variable que determina la posición en el eje horizontal y el valor de la otra variable que determina la posición en el eje vertical.

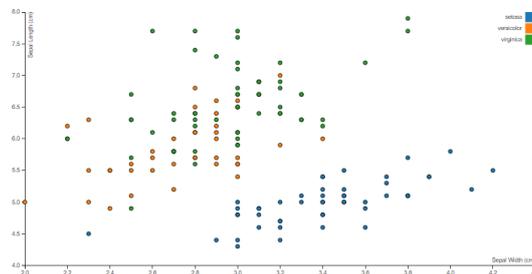


Figura 86. Scatterplot

Donut Chart

Un gráfico anillo es básicamente un gráfico circular con un área del centro recortada. Sin embargo, presenta como ventaja a diferencia de los gráficos circulares que el espacio en blanco central puede ser utilizado para mostrar información dentro de él. Algunas de las variaciones de este tipo de gráficos son Donut Múltiples, Animated Donut Chart with Labels o 3D Donut.



Figura 87. Donut Chart

Waterfall Chart

Este tipo de gráfico permite comprender el efecto acumulativo de los valores positivos o negativos introducidos secuencialmente. Estos valores intermedios pueden basarse en el tiempo o en la categoría.

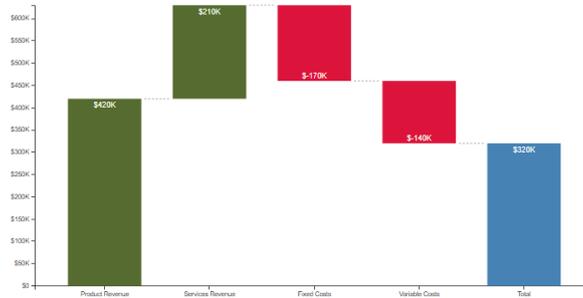


Figura 88. Waterfall Chart

Scree Plot

Un Scree Plot es una función decreciente que muestra la varianza explicada por cada factor en un análisis factorial. Se emplea para evaluar el número óptimo de factores a tener en cuenta, esto sucede alrededor del “codo”, punto en el que la meseta de la función no tiene en cuenta más factores.

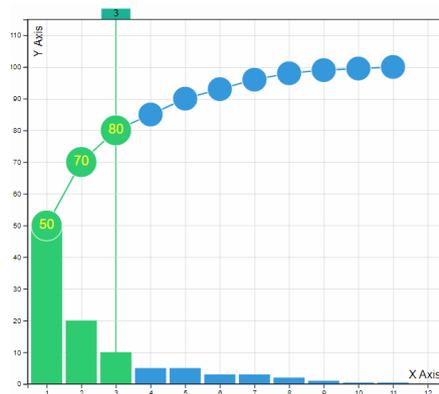


Figura 89. Scree Plot

En el presente trabajo se han puesto aquellos gráficos de D3.js con mayor relevancia, sin embargo, existen multitud de gráficos que no se han visto en este proyecto [8].

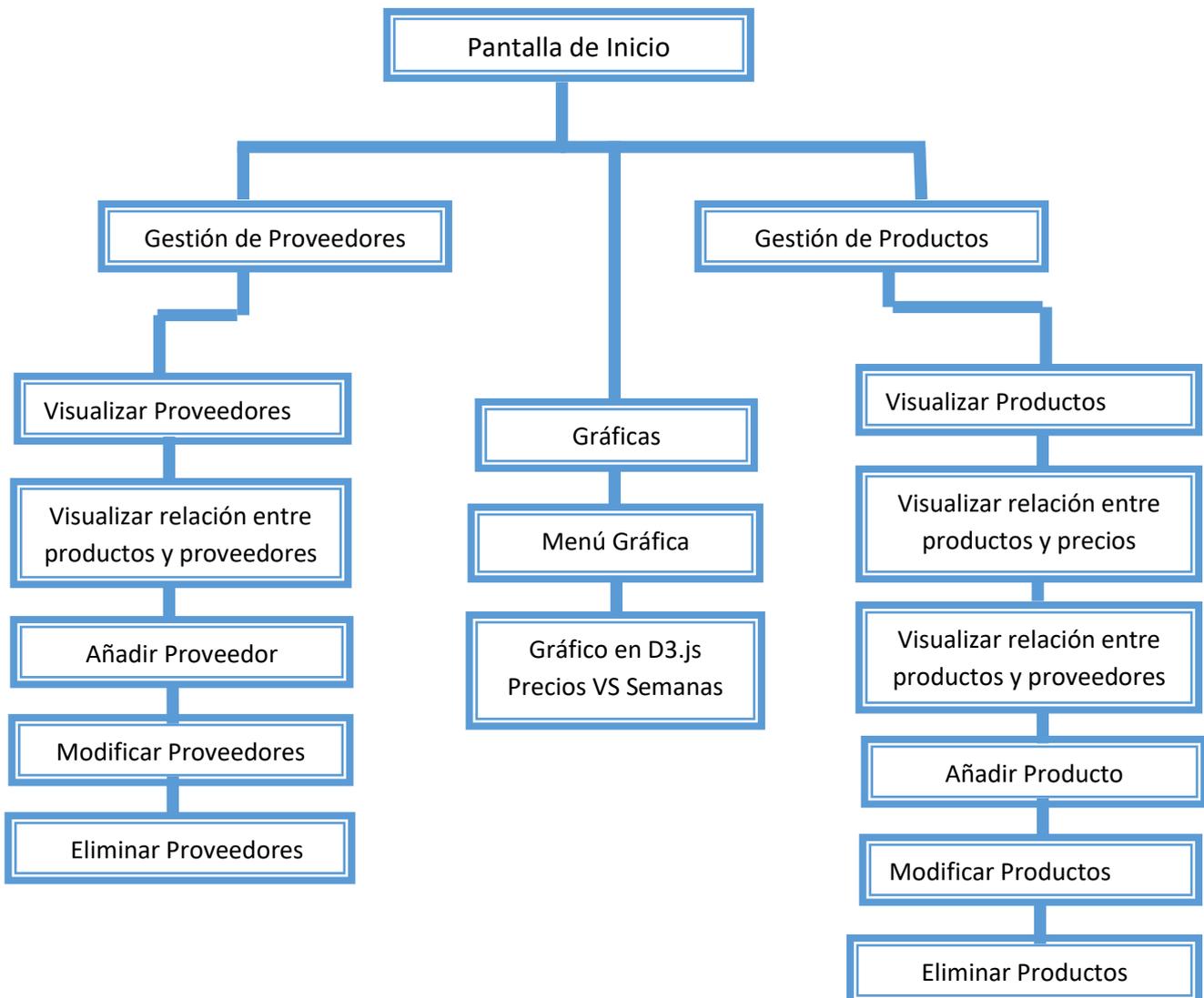
Capítulo V

Diagrama de pantallas y tecnología empleada

1. Diagrama de Pantallas

Un diagrama de pantalla representa una secuencia de procesos de software basados en usuario que se representan como una serie de interfaces gráficas de usuario.

Para poder mostrar todas las necesidades y requerimientos de funcionalidad, se ha diseñado diferentes diagramas que sean capaces de reflejar el funcionamiento según las pantallas que lo forman.



2. Tecnología empleada

En la realización del presente proyecto se han empleado distintos programas y lenguajes que se explicarán a continuación.

Para facilitar la explicación, se ha dividido la tecnología empleada en tres elementos principales: interfaz web, servidor de conexión y base de datos, de los cuales la interfaz web ya ha sido descrita brevemente en el capítulo 1.

Interfaz Web

Para la realización del diseño de los diagramas de pantallas que han sido mostrados en el apartado anterior, el lenguaje empleado es HTML 5.

HTML 5

HTML es un lenguaje marcado que se emplea para llevar a cabo el desarrollo de páginas de Internet. Las siglas HTML provienen de “Hyper Text Markup Language”, es decir, Lenguaje de Marcas de Hipertexto, que puede ser traducido como Lenguaje de Formato de Documentos para Hipertexto.

Este lenguaje surgió a partir de las etiquetas SGML (Standard Generalized Markup Language), las cuales se entienden como un sistema que permite ordenar y etiquetar diversos documentos dentro de una lista, debido a que no existen reglas para dicha organización, se dice que es un formato abierto.

HTML es un lenguaje muy simple y general que sirve para definir otros lenguajes que se encuentran relacionados con el formato de los documentos. De manera que HTML se encarga de desarrollar una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementando dicho texto como diversos objetos como fotografías, animaciones, etc.

HTML5 hace referencia a la quinta versión de HTML, este término representa dos conceptos diferentes:

- Es una nueva que versión contiene nuevos elementos, atributos y comportamientos.
- Abarca un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance.

Además, HTML5 especifica dos variantes de sintaxis para HTML, una clásica HTML (text/html) y una variante, XHTML (application/xhtml+xml) [9].

Servidor

Un servidor es un software capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia, es por ello que los servidores operan a

través de una arquitectura cliente-servidor, de manera que como ya se ha dicho atienden las peticiones de otros programas, los clientes, ofreciéndoles la posibilidad de compartir datos, información y recursos hardware y software.

Para llevar a cabo la selección del servidor se tuvieron en cuenta dos factores:

El primer factor es la compatibilidad del conjunto, de forma que se ha decidido simplificar los mecanismos de conexión tanto interfaz-servidor como servidor-base de datos.

El segundo factor es el conocimiento de los que dispone el desarrollador, en este caso abarca dos asignaturas del grado de Ingeniería de Organización Industrial, las cuales son “TICS para la gestión de la empresa” cursada en el tercer curso del grado, y “Fundamentos de informática” cursada en el primer curso del grado.

Es por ello que, en vista de los factores, el servidor seleccionado ha sido el Servidor PHP.

PHP

PHP (Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que además puede ser incrustado en HTML.

Este lenguaje se encuentra completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos. El código fuente escrito en PHP es invisible tanto para el navegador como para el cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar el resultado HTML al navegador.

Además, PHP permite incluir una interfaz de líneas de comandos que puede ser empleada en aplicaciones gráficas independientes.

Base de Datos

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que se necesiten en ese instante, dicho de otro modo, una base de datos es un sistema de archivos electrónicos.

Del mismo modo que para el servidor, es necesario seleccionar el programa de construcción de datos y por lo tanto tener en cuenta los conocimientos de los que dispone el desarrollador. De nuevo abarca dos asignaturas del grado de Ingeniería de Organización Industrial, las cuales son “TICS para la gestión de la empresa” cursada en el tercer curso del grado, y “Fundamentos de informática” cursada en el primer curso del grado. Por tanto, el programa seleccionado para la construcción de la base de datos es MariaDB.

MariaDB

MariaDB es un sistema de gestión de bases de datos (Database Management System, DBMS) derivado de MySQL con licencia GLP (General Public License) de código abierto para bases de datos relacionales, de manera que permite gestionar archivos llamados de bases de datos.

Ha sido desarrollado por Michael Widenius (fundador de MySQL), la fundación MariaDB y la comunidad de desarrolladores de software libre. La novedad que ofrece ese sistema de gestión de bases de datos es la introducción de dos motores de almacenamiento nuevos: Aria, que reemplaza con ventaja a MyISAM, y XtraDB, en sustitución de InnoDB.

Todo software que anteriormente emplease MySQL es capaz de seguir funcionando igualmente con MariaDB, es por ello, que se puede emplear en cualquier proyecto de nueva creación, así como intercambiar MySQL por MariaDB en todos aquellos proyectos que se encuentren ya en producción.

3. Otra tecnología empleada

CSS3

CSS son las siglas de “Cascading Style Sheets” es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación.

CSS fue desarrollado por W3C (World Wide Web Consortium) para permitir la separación de los contenidos de los documentos escritos en HTML, XML, XHTML, SVG o XUL de la presentación del documento con las hojas de estilo, incluyendo elementos tales como colores, fondos, márgenes, bordes, tipos de letra... modificando la apariencia de una página web de un modo sencillo.

JavaScript

JavaScript es un lenguaje de programación que surgió con el objetivo inicial de programar ciertos comportamientos sobre páginas web, respondiendo a la interacción del usuario y la realización de automatismos sencillos. Es decir, se podría decir que nació como un “lenguaje de scripting” del lado del cliente. Sin embargo, las necesidades de las aplicaciones web modernas y el HTML5 han provocado que el uso de JavaScript haya llegado a unos niveles de complejidad y prestaciones tan grandes como otros lenguajes de programación de primer nivel.

A JavaScript se le denomina “del lado del cliente” porque donde se ejecuta es en el navegador (cliente web), en contraposición a otros lenguajes como PHP que se ejecutan del “lado del servidor”, es decir, el cliente es el navegador que soporta la carga del procesamiento.

Con JavaScript se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos. Sin embargo, gracias a las API JavaScript del HTML5 se puede acceder a todo tipo de recursos adicionales como creación de gráficos basados en vectores y mapas de bits o flujos de datos con servidores [10].

D3.js

D3.js es una librería de JavaScript para producir, a partir, de datos, infogramas dinámicos e interactivos en navegadores web. No se profundizará más debido a que el capítulo 4 está dedicado íntegramente a esta librería.

Máquina Virtual cedida por la UVA

Una máquina virtual de sistema es un software que emula un ordenador completo, es decir, puede hacerse pasar por otro dispositivo de forma que se pueda ejecutar otro sistema operativo en su interior y por tanto tiene su propio disco duro, memoria, tarjeta gráfica y demás componentes de un hardware, aunque todos ellos son virtuales.

Una de las principales características que posee es que los procesos que se ejecutan se encuentran delimitados por los recursos y abstracciones proporcionados por ellos.

Para el sistema operativo que se ejecuta dentro de la máquina virtual, esta emulación es transparente e invisible. Para funcionar, una máquina virtual mapea los dispositivos virtuales que ofrece a su invitado con los dispositivos reales presentes en la máquina física.

Capítulo VI

HTML y PHP

1. Introducción HTML5

HTML5 (Hyper Text Markup Language) es la quinta versión del lenguaje de marcado estándar que se emplea para la web. En la actualidad es uno de los principales lenguajes de marcado más empleados en el mundo porque permite crear la estructura de una página web, es decir, texto, imágenes y material multimedia pueden mostrarse correctamente gracias a HTML5.

Para una mayor comprensión de la definición, se definirá el término lenguaje de marcado.

Un lenguaje de marcado hace referencia a aquellos lenguajes que utilizan etiquetas, las cuales ya se encuentran predefinidas dentro del lenguaje respectivo y contienen información que ayudan a leer el texto. Es decir, tanto para las plataformas como para los desarrolladores, las etiquetas contienen información adicional de la estructura del texto.

La principal diferencia respecto a los lenguajes de programación es que estos últimos poseen funciones aritméticas o variables, mientras que los lenguajes de marcado no.

La estructura básica de un documento HTML5 es la siguiente:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8"/>
    <title>Título de la Página
  </title>
  </head>
  <body>
    Aquí va el contenido de la página
  </body>
</html>
```

Figura 90. Estructura básica documento HTML5

La declaración `<!DOCTYPE html>` es el primer elemento que debe aparecer en un documento HTML ya que informa al navegador con el que se abre la página web acerca del lenguaje y la versión del documento.

La etiqueta `<html>` y `</html>` es la etiqueta que se pone inmediatamente después de la declaración DOCTYPE, comprende e indica todo aquello dentro del documento que es lenguaje HTML.

La etiqueta `<head>` y `</head>` se encuentra localizada en la cabecera del documento y debe contener obligatoriamente el título de la página, es decir, la etiqueta `<title>` y `</title>`. Puede contener otros elementos que no serán visualizados en la página pero que si aportaran información al navegador acerca de ella. Entre otros, los elementos que pueden localizarse dentro del head son la etiqueta `<style>` y `</style>`, con reglas CSS internas. Además, también puede contener metadatos (`<meta/>`) que por lo general se emplean para especificar la descripción de la página, la última modificación, etc.

La etiqueta `<title>` y `</title>` es el título del documento HTML y debe especificarse en cada una de las páginas que componen un sitio web. No se debe confundir con el nombre de un dominio ni con el nombre dado al documento al guardarlo.

Por último, la etiqueta `<body>` y `</body>` representa el contenido de un documento HTML, siendo parte de la estructura básica de un documento. Solo puede haber un elemento `<body>` en cada documento [11].

2. Pantallas HTML5

Las pantallas diseñadas en HTML5 tienen como objetivo facilitar un entorno visual sencillo que permita la comunicación con el sistema operativo de un ordenador.

Las diferentes pantallas que se van a encontrar son las siguientes:

- Principal.html
- Menu.html
- Menu_proveedores.html
- Menu_productos.html
- Anadir_proveedor.html
- Anadir_producto.html
- Eliminar_producto.html
- Eliminar_proveedor.html
- Modificar_producto.html
- Modificar_proveedor.html
- Graficas.html

El archivo Graficas.html no se explicará en el presente capítulo, debido que el capítulo VIII, está íntegramente dedicado a las gráficas creadas en D3.js.

A continuación, se irá una a una explicando cada una de las pantallas HTML con su funcionalidad.

Principal.html

Es la pantalla de acceso al sistema. Está basado en la validación del acceso a través del usuario y la contraseña, de forma que cada usuario que quiera acceder al sistema con su nombre de usuario y contraseña tendrá acceso al mismo si está registrado.



The screenshot shows a web page titled "EMPRESA" in a dark blue header. Below the header is a white background with a blue-bordered box containing a login form. The form has two input fields: "Usuario:" and "Contraseña:". Below the fields is a blue button labeled "Enviar".

Figura 91. Principal.html

Menu.html

Una vez que el usuario ha accedido al sistema, se le plantea la posibilidad de o bien acceder a la “gestión de proveedores”, que a su vez contendrá otro menú que se explicará más adelante o acceder a la “gestión de productos”, que en este caso también le permitirá al usuario acceder a otro menú donde podrá elegir la opción que desee.



The screenshot shows a web page titled "Menú" in a dark blue header. Below the header is a white background with a blue-bordered box containing a menu. The menu has three options: "Gestión de proveedores", "Gestión de productos", and "Gráficas". Below the menu is a blue button labeled "Atrás".

Figura 92. Menu.html

Menu_proveedores.html

Si el usuario ha decidido que desea acceder a la “gestión de proveedores” en este caso se encontrará con otro menú en el que tendrá la posibilidad de “visualizar proveedores”, “visualizar relación entre productos y proveedores”, “añadir proveedor”, “modificar proveedor” o “eliminar proveedor”, estos tres últimos referidos a proveedores de la base de datos.



Figura 93. Menu_proveedores.html

Menu_productos.html

Si por el contrario el usuario ha decidido que desea acceder a la “gestión de productos” en este caso se encontrará con otro menú en el que tendrá la posibilidad de “visualizar productos”, “visualizar relación entre productos y precios”, “visualizar relación entre productos y precios”, “añadir producto”, “modificar productos” o “eliminar productos”, estos tres últimos referidos a productos de la base de datos.

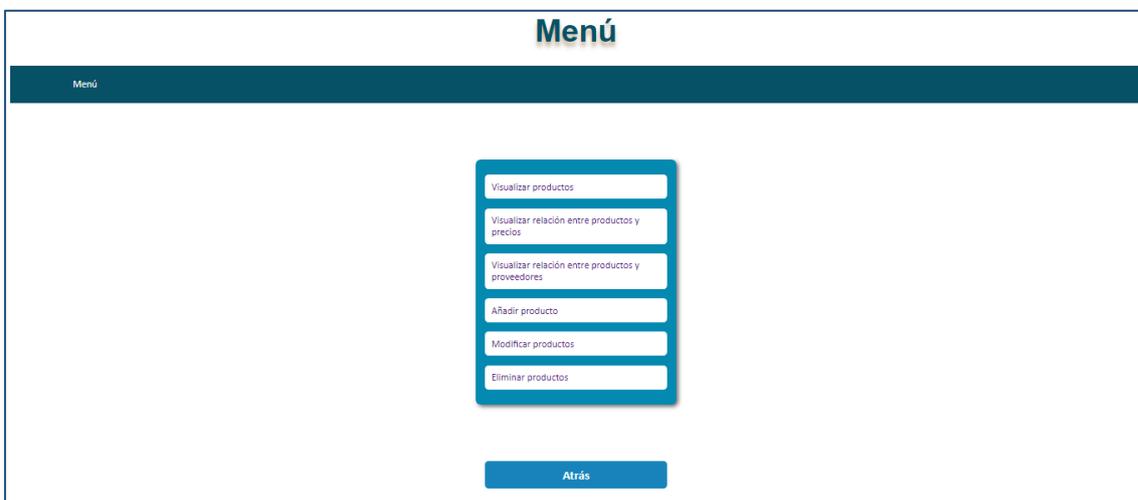
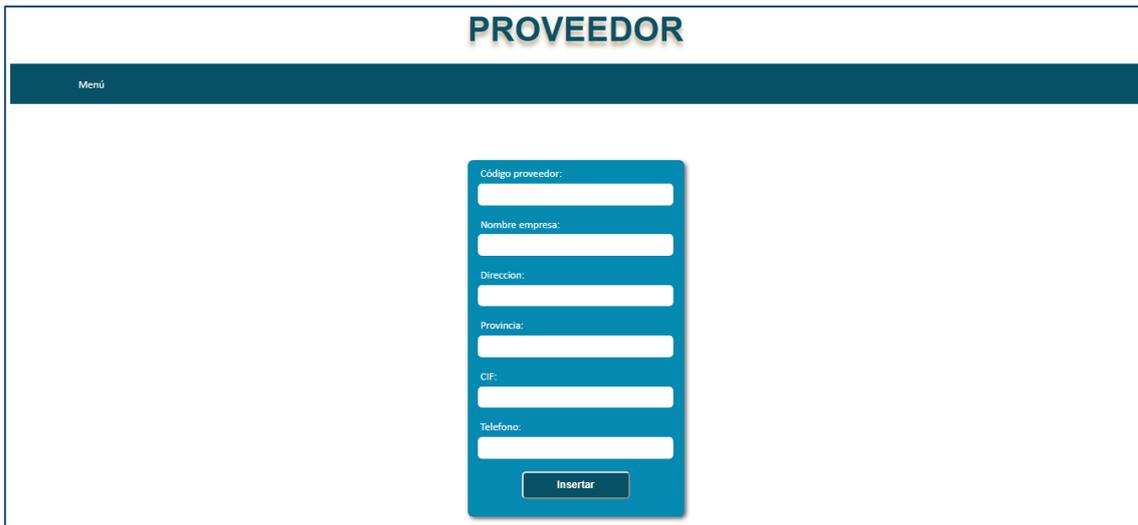


Figura 94. Menu_productos.html

Añadir_proveedor.html

Esta pantalla permite al usuario añadir un nuevo proveedor a la base de datos. Para ello debe introducir diferentes datos como el nuevo código del proveedor, el nombre de la empresa, la dirección de la empresa, la provincia donde se encuentra localizada la empresa, el CIF de la empresa y, por último, el teléfono de la empresa.



The screenshot shows a web page titled 'PROVEEDOR'. At the top, there is a dark blue navigation bar with the word 'Menú' on the left. Below this, the main content area is white. In the center, there is a blue-bordered form box. Inside this box, there are several input fields, each with a label above it: 'Código proveedor:', 'Nombre empresa:', 'Direccion:', 'Provincia:', 'CIF:', and 'Telefono:'. Each label is followed by a white input field. At the bottom of the form box, there is a dark blue button with the text 'Insertar' in white.

Figura 95. Anadir_proveedor.html

Eliminar_proveedor.html

Esta pantalla permite al usuario eliminar un nuevo proveedor de la base de datos. Para ello se debe introducir el código del proveedor que se desea eliminar de la base de datos.



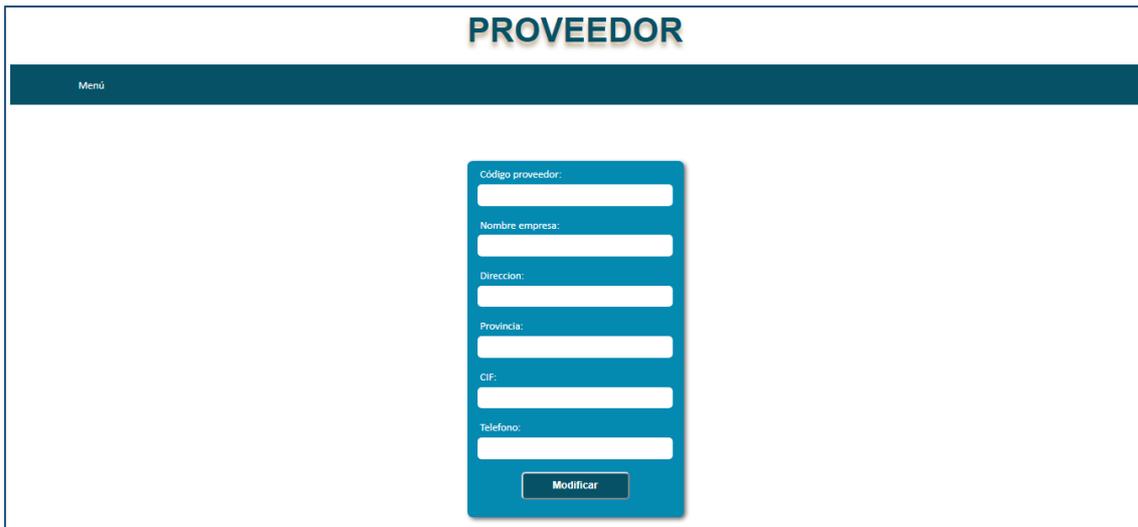
The screenshot shows a web page titled 'PROVEEDOR'. At the top, there is a dark blue navigation bar with the word 'Menú' on the left. Below this, the main content area is white. In the center, there is a blue-bordered form box. Inside this box, there is a single input field with the label 'Código proveedor:' above it. Below the input field, there is a dark blue button with the text 'Eliminar' in white.

Figura 96. Eliminar_proveedor.html

Modificar_proveedor.html

Esta pantalla permite al usuario modificar los datos de un proveedor en la base de datos. Para ello debe modificar diferentes datos como el código del proveedor, el nombre de la empresa, la dirección de la empresa, la provincia donde se encuentra localizada la empresa, el CIF de la empresa y, por último, el teléfono de la empresa.

Aquello que se desee que se mantenga igual dentro de los datos, se dejará vacío de manera que solo se introducirán los cambios nuevos. Lo que es imprescindible de introducir es el código.



The screenshot shows a web page titled "PROVEEDOR". At the top, there is a dark blue header with the word "Menú" on the left. Below the header, the main content area is white. In the center, there is a blue-bordered form box. Inside this box, there are several input fields: "Código proveedor:", "Nombre empresa:", "Direccion:", "Provincia:", "CIF:", and "Telefono:". Each field has a white input area. At the bottom of the form box is a dark blue button labeled "Modificar".

Figura 97. Modificar_proveedor.html

Anadir_producto.html

Esta pantalla permite al usuario añadir un nuevo producto a la base de datos. Para ello se debe introducir diferentes datos como el nuevo código del producto y el nombre del producto.



The screenshot shows a web page titled "PRODUCTO". At the top, there is a dark blue header with the word "Menú" on the left. Below the header, the main content area is white. In the center, there is a blue-bordered form box. Inside this box, there are two input fields: "Código producto:" and "Nombre producto:". Each field has a white input area. At the bottom of the form box is a dark blue button labeled "Insertar".

Figura 98. Anadir_producto.html

Eliminar_producto.html

Esta pantalla permite al usuario eliminar un nuevo producto de la base de datos. Para ello se debe introducir el código del producto que se desea eliminar de la base de datos.



The screenshot shows a web page titled "PRODUCTO". At the top, there is a dark blue header with the word "Menú" on the left. Below the header, the main content area is white. In the center, there is a blue-bordered form box. Inside this box, there is one input field: "Código producto:". The field has a white input area. At the bottom of the form box is a dark blue button labeled "Eliminar".

Figura 99. Eliminar_producto.html

Modificar_producto.html

Esta pantalla permite al usuario modificar los datos de un proveedor en la base de datos. Para ello debe modificar diferentes datos como el código del producto y el nombre del producto.

Aquello que se desee que se mantenga igual dentro de los datos, se dejará vacío de manera que solo se introducirán los cambios nuevos. Lo que es imprescindible de meter es el código.



Figura 100. Modificar_producto.html

3. Análisis pantallas HTML5

A continuación, se va a ir analizando una a una las diferentes pantallas creadas en HTML.

Principal.html

En primer lugar, se encuentra la etiqueta `<html>` que como ya se ha explicado anteriormente informa al navegador con el que se abre la página web acerca del lenguaje y la versión del documento.

Le sigue la etiqueta `<head>`, que engloba:

- El título del documento en HTML `<title>`.
- El enlace al archivo CSS que contiene el estilo.
- Y, por último, `<meta http-equiv="Content-Type" content="text/html"; charset="utf-8"/>` que indica al servidor que el contenido de la página es código HTML y su codificación de caracteres es UTF-8.

Se cierra el encabezado con `</head>`.

Le sigue el cuerpo `<body>` que engloba:

- `<div class="titulo">EMPRESA</div>` a través del cual se ha puesto como título general “EMPRESA”.
- `<div class="menu_inicio">` que indica el estilo del CSS que se desea para el contenedor.

- Seguimiento del `<div class="contenedor">` que lo que permite al usuario es introducir su nombre de usuario y contraseña. Cabe destacar que cuando el usuario rellena el formulario y da a "Insertar", se llama a la página `usuario.php` a través de `<form action="usuario.php" method="POST">` de manera que se pasan los parámetros necesarios para comprobar si el usuario puede acceder a la web. Finalmente, para cerrar el cuerpo se encuentra `</body>` y para cerrar el código html `</html>`

```

<html>
  <head>
    <title>Login</title>
    <link rel="stylesheet" type="text/css" href="estilo.css">
    <meta http-equiv="Content-Type" content="text/html"; charset="utf-8"/>
  </head>

  <body>
    <div class="titulo">EMPRESA</div>
    <div id="menu_inicio">--</div>

    <div class="contenedor">
      <form action="usuario.php" method="POST">
        <label>Usuario: </label>
        <input type="text" name="username"/><br>
        <label>Contraseña: </label>
        <input type="password" name="password"/><br><br>
        <input id="boton-login" type="submit" value="Enviar"/>
      </form>
    </div>
  </body>
</html>

```

Figura 101. Código Principal.html

Menu.html

El código `Menu.html` es exactamente igual que el código de `principal.html`, la diferencia reside en que `menu.html` incluye:

- `Atrás` que permite al usuario retroceder a la página anterior pulsando el botón atrás.
- `...` que permite acceder a `Menu_proveedores.html`.
- `...` que permite acceder a `Menu_productos.html`.
- Y `...` que permite acceder a `Graficas.php`.

`Menu_proveedores.html` y `Menu_productos.html` es exactamente igual que `Menu.html`.

```

<head>
  <title>Menu</title>
  <link rel="stylesheet" type="text/css" href="estilo.css">
  <meta http-equiv="Content-Type" content="text/html"; charset="utf-8"/>
</head>
<body>
  <div class="titulo">Menú</div>
  <div id="menu_inicio">-</div>

  <div class="vertical-menu">
    <div class="contenedor">
      <div class="element"><a class="link" href="Menu_proveedores.html">Gestión de proveedores</a></div>
      <div class="element"><a class="link" href="Menu_productos.html">Gestión de productos</a></div>
      <div class="element"><a class="link" href="Menu_grafica.php">Gráficas</a></div>
    </div>
  </div>

  <a class="boton-atras" href="principal.html">Atrás</a>
</body>

```

Figura 102. Código Menu.html

Anadir_proveedor.html

```

<html>
  <head>
    <title>Añadir proveedor</title>
    <link rel="stylesheet" type="text/css" href="estilo.css">
    <meta http-equiv="Content-Type" content="text/html"; charset="utf-8">
  </head>

  <body>
    <div class="titulo">PROVEEDOR</div>
    <div class="menu_simple">
      <ul>
        <li><a href="menu.html">Menú</a></li>
      </ul>
      <br />
    </div>

    <div class="contenedor">
      <form action="anadir_proveedor.php" method="POST">
        <label>Código proveedor: </label>
        <input type="text" name="codigo"/><br>
        <label>Nombre empresa: </label>
        <input type="text" name="nombre"/><br>
        <label>Direccion: </label>
        <input type="text" name="direccion"/><br>
        <label>Provincia: </label>
        <input type="text" name="provincia"/><br>
        <label>CIF: </label>
        <input type="text" name="cif"/><br>
        <label>Telefono: </label>
        <input type="text" name="telefono"/><br>
        <input id="boton-login" type="submit" value="Insertar"/>
      </form>
    </div>
  </body>
</html>

```

Figura 103. Código Anadir_proveedor.html

El código `Anadir_proveedor.html` es exactamente igual que el código de `Eliminar_proveedor.html` y `Modificar_proveedor.html`, y también que `Anadir_producto`, `Eliminar_producto.html` y `Modificar_producto.html` por lo que no se explicarán dichos códigos.

4. Introducción PHP

PHP (Hypertext Pre-Processor) es un lenguaje de código abierto, es decir, es de uso libre y gratuito para que todos los programadores puedan utilizarlo sin límite, que permite programar scripts del lado del servidor que se incrusta dentro del código HTML, y por tanto, un mismo archivo puede combinar código PHP con código HTML.

El hecho de que sea un lenguaje del lado del servidor implica que se ejecuta en el servidor web, justo antes de que se envíe la página a través de internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras muchas tareas para llevar a cabo la creación de la página web final que el cliente podrá visualizar. De forma que el cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP.

Además, este tipo de lenguaje permite generar páginas web dinámicas, es decir, páginas web cuyo contenido no siempre es el mismo.

La estructura básica de PHP es la siguiente:

```
<?php
¿>
```

Como se puede ver es muy sencilla ya que solo es necesario `<?php` para abrir y `?>` para cerrar. Sin embargo, lo que realmente hay que entender es el código inmerso en `<?php ¿>`, por lo que a continuación se irá analizando pantalla a pantalla de php para poder entender la estructura.

5. Pantallas PHP

Las diferentes pantallas que se van a encontrar son las siguientes:

- `productos.php`
- `productos_precios.php`
- `productos_vendedores.php`
- `proveedores.php`
- `usuarios.php`
- `menu_productos.php`
- `anadir_proveedor.php`
- `anadir_producto.php`
- `eliminar_producto.php`

- eliminar_proveedor.php
- modificar_producto.php
- modificar_proveedor.php
- menu_grafica.php

El archivo menu_grafica.php no se explicará en el presente capítulo, debido que el capítulo VIII, está íntegramente dedicado a las gráficas creadas en D3.js.

A continuación, se irá algunas de las pantallas PHP con su funcionalidad.

Proveedores.php

Esta pantalla permite al usuario visualizar los datos correspondientes a los diferentes proveedores con lo que cuenta la empresa. Entre los datos se encuentran el código, nombre, dirección, provincia, CIF y teléfono correspondiente a cada uno de los proveedores.

Proveedores					
Código	Nombre	Dirección	Provincia	CIF	Teléfono
1	Navarro Montes	calle tratado de tordesillas n98	valladolid	71171119x	674329064
2	Hermanos Gil	calle bartolome de las casas n21	valladolid	71648564x	645709327
3	Maya Blue	calle valentin calderon n17	valladolid	71753495x	692875934
4	García Rey	calle primera universidad n43	valladolid	71457992x	601294390

[Atrás](#)

Figura 104. Proveedores.php

Productos_vendedores.php

Esta pantalla permite al usuario visualizar cuales son las fechas en las que los proveedores suministran a la empresa los pedidos, y por tanto, podrá visualizar que producto llega a la empresa en una fecha y cuál es su proveedor.

Relación productos y vendedores		
Producto	Proveedor	Fecha
AGUJA	Maya Blue	2018-01-03
AGUJA	Navarro Montes	2018-01-03
AGUJA	Hermanos Gil	2018-01-03
AGUJA	Maya Blue	2018-01-10
AGUJA	Navarro Montes	2018-01-10
AGUJA	Hermanos Gil	2018-01-10
AGUJA	Maya Blue	2018-01-17
AGUJA	Navarro Montes	2018-01-17
AGUJA	Hermanos Gil	2018-01-17
AGUJA	Maya Blue	2018-01-24
AGUJA	Navarro Montes	2018-01-24
AGUJA	Hermanos Gil	2018-01-24

Figura 105. Productos_vendedores.php

Productos.php

Esta pantalla permite al usuario visualizar cuales son los productos que pone a la venta la empresa, así como el código de cada producto.

Gráficas	
Código	Nombre del producto
62	SALCHICHA de POLLO 2 kilos
96	MAGRO Limpieza de chuletero 7525 min 25
2037	SOLOMILLOS(VACIO) 5 unidades
2055	CARRILLERA DE CERDO VACIO 1KG
2115	CONTRA extratierna 2 trozos vacio
2124	LOMO EXTRATIERNO (VACIO)
2126	SOLOMILLO PR. EXTRATIERNO (VACIO)
2131	CHULET sin cabec sin solomillo Extratier
2132	LOMO ADOBADO EXTRATIERNO
2142	JAMON ADOBADO EXTRATIERNO Deshues

Figura 106. Productos.php

Productos_precios.php

Esta pantalla permite visualizar el nombre del producto, así como su precio anterior, su precio actual, la variación de precio y la cantidad a lo largo de las 52 semanas que componen el año.

Relación productos y precios					
Producto	Precio anterior	Variación	Precio actual	Semanas	Cantidad (kg)
SALCHICHA de POLLO 2 kilos	4.05	0.05	4.10	S1	28
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S10	28
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S11	27
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S12	20
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S13	26
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S14	20
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S15	20
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S16	22
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S17	22
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S18	20
SALCHICHA de POLLO 2 kilos	4.10	0.05	4.15	S19	29
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S2	25

Figura 107. Productos_precios.php

6. Análisis pantallas PHP

A continuación, irán una a una explicando cada una de las pantallas PHP con su funcionalidad.

Anadir_proveedor.php

Para crear la conexión con el servidor de la base de datos se emplea la función `mysqli_connect()` y los datos o variables que se acaban de crear. Para ello se crea

una variable `$conexion` que contendrá la conexión de forma que queda:

```
$conexion = mysqli_connect($host_db, $user_db, $pass_db, $db_name);
```

A continuación, le sigue el condicional que devuelve TRUE en caso de conexión satisfactoria con MariaDB o FALSE en caso de error.

```
if (!$conexion) {
    die("La conexión falló: " . mysqli_error());
}
else
{
    echo "Conexión satisfactoria";
}
```

El siguiente paso es llevar a cabo la recepción de información del usuario a través del método POST, y por tanto, la información se transmite mediante protocolo HTTP siendo los datos invisibles al usuarios.

Una vez transmitida la información, el siguiente objetivo es ingresar datos en tablas a través del comando INSERT: `$sql = "INSERT INTO $tbl_name ()";`

Tras el ingreso de los datos, se encuentra dos condicionales, el primero que devuelve el resultado de la operación en caso de que todos los datos introducidos por el usuario sean NOT NULL, o que devuelva error en caso contrario. Y el segundo condicional devolverá “Se ha insertado en la base de datos” en caso de éxito, y “No se ha insertado” en caso de fallo.

Finalmente, para cerrar la conexión de la base se utiliza la función `mysqli_close()`.

```
<?php
$host_db = "localhost";
$user_db = "root";
$pass_db = "";
$db_name = "tfg_db";
$tbl_name = "proveedores";

$conexion = mysqli_connect($host_db, $user_db, $pass_db, $db_name);
$conexion->set_charset("utf8");

if (!$conexion) {
    die("La conexión falló: " . mysqli_error());
}
else
{
    echo "Conexión satisfactoria";
}

$codigo = $_POST['codigo'];
$nombre = $_POST['nombre'];
$direccion = $_POST['direccion'];
$provincia = $_POST['provincia'];
$cif = $_POST['cif'];
$telefono = $_POST['telefono'];

$sql = "INSERT INTO $tbl_name (id_proveedor, nomempresa_proveedor, direccion_proveedor, provincia_proveedor, cif_proveedor,
telef_proveedor) VALUES ('$codigo', '$nombre', '$direccion', '$provincia', '$cif', '$telefono)";

if($codigo != null && $nombre != null && $direccion != null && $provincia != null && $cif != null && $telefono != null)
{
    $result = mysqli_query($conexion, $sql);
}
else
{
    echo "<script>alert('Error. Faltan datos.'); window.location.href=\"Menu_proveedores.html\"</script>";
}
```

Figura 108. Código Anadir_Proveedor.php Parte 1

```
if ($result) {
    echo "<script>alert('Se ha insertado en la base de datos.');
```

Figura 109. Código Anadir_Proveedor.php Parte 2

El código **Anadir_producto.php** es exactamente igual que el código de **Anadir_proveedor.php**. Así como **Eliminar_producto.php**, **Eliminar_proveedor.php**, **Modificar_producto.php** y **Modificar_proveedor.php** que presentan también el mismo código a diferencia de que para los dos primeros emplean el comando DELETE se permite eliminar registros existentes en una tabla, en el caso de **eliminar_producto.php** sería: `$sql = "DELETE FROM $tbl_name WHERE id_producto = '$codigo'";`. Para los dos últimos de nuevo también sería el mismo código a diferencia de que en este caso se emplearía el comando UPDATE para actualizar los nuevos valores. El resto del código es exactamente igual para los 4 casos por lo que no se explicarán.

Productos.php

El código comienza con dos cadenas de caracteres en php que muestran el estilo que tendrá la página con php, se representa a través de “echo”.

Para crear la conexión con el servidor de la base de datos se emplea la función `mysqli_connect()` y los datos o variables que se acaban de crear. Para ello se crea una variable `$conexion` que contendrá la conexión de forma que queda:

```
$conexion = mysqli_connect($host_db, $user_db, $pass_db, $db_name);
```

A continuación, le sigue el condicional que devuelve TRUE en caso de conexión satisfactoria con MariaDB o FALSE en caso de error.

```
if (!$conexion) {
    die("La conexión falló: " . mysqli_error());
}
else
{
    echo "Conexión satisfactoria";
}
```

Le sigue otro condicional que devuelve la tabla de productos en caso de éxito y “No existe ningún producto en la base de datos” en caso de que no se encuentre el producto. Además, dentro del condicional se encuentra un bucle while que proporcionará estilo a la tabla.

Finalmente, para cerrar la conexión de la base se utiliza la función `mysqli_close()`.

```

<?php
echo "<div style='font-family: Arial, sans-serif; text-align: center; font-size: 50px; font-weight: bold;
color: #075166; text-shadow: 0 8px 9px #c4b59d, 0px -2px 1px #fff; border-radius: 20px; height: 10%;'>Productos</div>";
echo "<div style='background-color: #075166; height: 5%; width: 100%;'> </div>";

$host_db = "localhost";
$user_db = "root";
$pass_db = "";
$db_name = "tfg_db";
$tbl_name = "producto";

$conexion = mysqli_connect($host_db, $user_db, $pass_db, $db_name);

if (!$conexion) {
    die("La conexión falló: " . mysqli_error());
}

$sql = "SELECT * FROM $tbl_name order by id_producto";
$result = mysqli_query($conexion, $sql);

if (mysqli_num_rows($result) != 0) {
    echo "<table style = 'margin: 0 auto; display:block; margin-top: 3%; font-family: calibri, sans-serif;'>";
    echo "<tr>";
    echo "<th style = 'padding-bottom: 5%;'>Código</th>";
    echo "<th style='text-align: left; padding-left: 10%; padding-right: 10%; padding-bottom: 5%;'>Nombre del producto</th>";
    echo "</tr>";
    while ($row = mysqli_fetch_row($result)){
        echo "<tr>";
        echo "<td style='text-align: left; padding-right: 10%;'>".$row[0]."</td>";
        echo "<td style='text-align: left; padding-left: 10%; padding-right: 10%;'>".$row[1]."</td>";
        echo "</tr>";
    }
    echo "</table>";
} else {
    echo "<script>alert('No existe ningún producto en la base de datos.');" window.location.href="principal.html"</script>";
}

mysqli_close($conexion);

echo "<a style='font-family: calibri, sans-serif; margin-bottom: 10%; text-decoration: none; padding:
10px; font-weight: 600; font-size: 20px; color: #ffffff; background-color: #1883ba; border-radius: 6px;
margin: 0 auto; display: block; width: 15%; text-align: center; margin-top:5%;' href='menu.html'>Atrás</a>"
?>

```

Figura 110. Código productos.php

El código productos.php es exactamente igual que el código **productos_precios.php**, **productos_vendedores.php** y **proveedores.php**, por lo que no se explicarán dichos códigos.

Usuario.php

El inicio de este código es exactamente igual que para **anadir_proveedor.php** y **productos.php**, por lo que no se explicará.

El siguiente paso es llevar a cabo la transmisión de información del usuario a través del método POST, y por tanto, la información se transmite mediante protocolo HTTP siendo los datos invisibles al usuarios.

Una vez transmitida la información, el siguiente objetivo es comprobar si existe el usuario en la tabla a través del comando SELECT: `$sql = "SELECT * FROM $tbl_name ()";`

A continuación, encontramos un condicional que permite entrar al usuario en la página web en caso de que el usuario y la contraseña sean correctos, y además sea antes del tiempo en que la sesión expira que es tras los 5 minutos, o obtener "Username o Password están incorrectos" en caso de error.

Finalmente, para cerrar la conexión de la base se utiliza la función `mysqli_close()`.

```
<?php
$host_db = "localhost";
$user_db = "root";
$pass_db = "";
$db_name = "tfg_db";
$tbl_name = "usuarios";

$conexion = mysqli_connect($host_db, $user_db, $pass_db, $db_name);
$conexion->set_charset("utf8");

if (!$conexion) {
    die("La conexión falló: " . mysqli_error());
}
else
{
    echo "Conexión satisfactoria";
}

$username = $_POST['username'];
$password = $_POST['password'];

$sql = "SELECT * FROM $tbl_name WHERE nom_usuario = '$username' and password_usuario = '$password'";
$result = mysqli_query($conexion, $sql);

if (mysqli_num_rows($result) > 0) {
    $_SESSION['loggedin'] = true;
    $_SESSION['username'] = $username;
    $_SESSION['start'] = time();
    $_SESSION['expire'] = $_SESSION['start'] + (5 * 60);

    header("Location: menu.html");
}
else {
    echo "<script>alert('Username o Password estan incorrectos.');
```

Figura 111. Código usuario.php

Capítulo VII

Diseño de la base de datos

Una base de datos es un conjunto de datos informativos organizados de modo que sean fácilmente accesibles, gestionados y actualizados. Estos bancos de información comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto.

Una base de datos bien estructurada proporciona un ahorro de espacio en el disco eliminando aquellos datos redundantes, mantiene la precisión e integridad de los datos y ofrece el acceso a los datos de formas útiles.

El diseño de una base de datos útil y eficiente requiere seguir el proceso adecuado, incluidas las siguientes etapas:

1. Análisis de los requisitos o identificación del propósito de la base de datos.
2. Organización de los datos en tablas.
3. Especificación de las claves primarias y análisis de las relaciones.
4. Normalización para estandarizar las tablas.

A su vez en el diseño de la base de datos se divide en tres partes secuenciales que modelan la realidad con diferente forma de caracterización y nivel de detalle. Las cuales son diseño Conceptual, diseño Lógico y diseño Físico.

El diseño Conceptual consiste en realizar una descripción de alto nivel del contenido de información de la base de datos, independientemente del sistema de gestión de bases de datos que se vaya a emplear. El principal objetivo es describir el contenido de información de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar dicha información. En esta fase se obtendrá un esquema completo denominado Modelo Conceptual de Datos (MCD).

El diseño Lógico parte del resultado del diseño Conceptual y permite obtener una descripción de la estructura de la base de datos en términos de estructura de datos que puede procesar un tipo de sistema de gestión de bases de datos (SGBD). Es por tanto que consiste en realizar una descripción de la estructura de la base de datos según el modelo del sistema de gestión de bases de datos que vaya a ser utilizado. El principal objetivo es obtener una representación que emplee de la manera más eficiente los recursos disponibles para estructurar datos y modelar restricciones, a esta representación se la llama Modelo Lógico de Datos (MLD).

Por último, el diseño Físico parte del Lógico y da como resultado una descripción de la implantación de una base de datos en la memoria secundaria: estructuras de

almacenamiento y métodos empleados para tener un acceso efectivo a los datos. El objetivo es conseguir una mayor eficiencia y se debe tener en cuenta aspectos concretos del SGBD sobre el que se vaya a implementar. Se obtiene a partir de ambos modelos descritos anteriormente, MCD y MLD. Por regla general el diseño Físico es transparente para el usuario, aunque conocer cómo se implementa ayuda a optimizar el rendimiento y la escalabilidad del sistema [12].

1. Modelo Conceptual de Datos

El Modelo Conceptual de datos, también denominado modelo entidad-relación, como ya se ha comentado anteriormente este modelo es sólo y exclusivamente un método del que se dispone para diseñar un esquema que posteriormente será implementado en el SGBD. Este modelo se representa a través de diagramas y está formado por varios elementos entre los que se encuentran las entidades, los atributos y las relaciones.

- **Entidades:** Objeto del mundo real sobre el que se quiere almacenar la información y se distingue de otros objetos de acuerdo con sus características llamadas atributos. Las entidades pueden ser concretas o abstractas. De manera que, si la entidad es un objeto con existencia física, entonces se estaría hablando de entidad concreta, a diferencia de un objeto con existencia conceptual que en este caso se estaría hablando de entidad abstracta.
- **Atributos:** Son las propiedades que describen a cada entidad en un conjunto de entidades. Un conjunto de entidades dentro de una entidad tiene valores específicos asignados para cada uno de sus atributos, de esta manera, es posible su identificación unívoca.
Se puede hablar de atributos identificativos, como aquellos que permiten diferenciar a una entidad de otra diferente. Para cada atributo existe un dominio del mismo, el cual hace referencia al tipo de datos que será almacenado o a restricciones en los valores que el atributo puede tomar (cadenas de caracteres, números, etc.).
- **Relación:** Es una asociación o relación matemática entre varias entidades o consigo mismas. Cada entidad interviene en una relación con una determinada cardinalidad. La cardinalidad (número de elementos de una entidad que pueden asociarse a un elemento de la otra entidad relacionada) se representa mediante una pareja de datos, en minúsculas, de la forma (cardinalidad mínima, cardinalidad máxima), asociada a cada una de las entidades que intervienen en la relación. En este modelo se emplearán verbos que conecten las entidades como relación.

De forma que una vez que se ha entendido la diferencia entre los tres principales elementos del modelo entidad-relación, se va a proceder a la construcción del mismo.

En primer lugar, se mostrará el modelo E-R sin atributos ni cardinalidad de las relaciones.

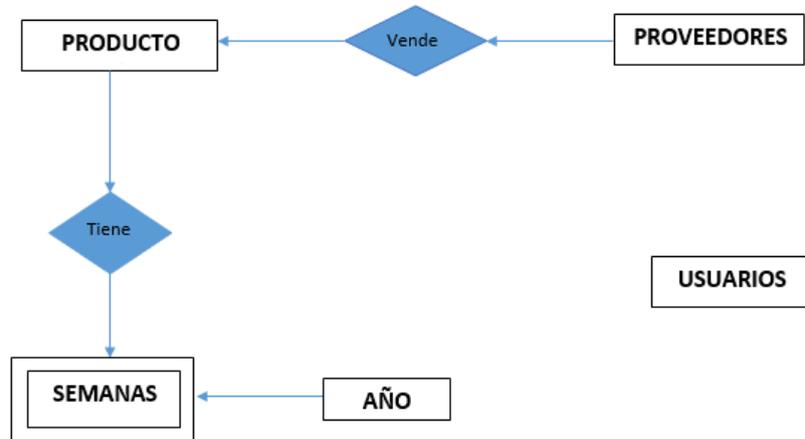


Figura 112. Modelo E-R sin atributos ni cardinalidad

En segundo lugar, se mostrará de nuevo el modelo E-R, pero en este caso con sus atributos.

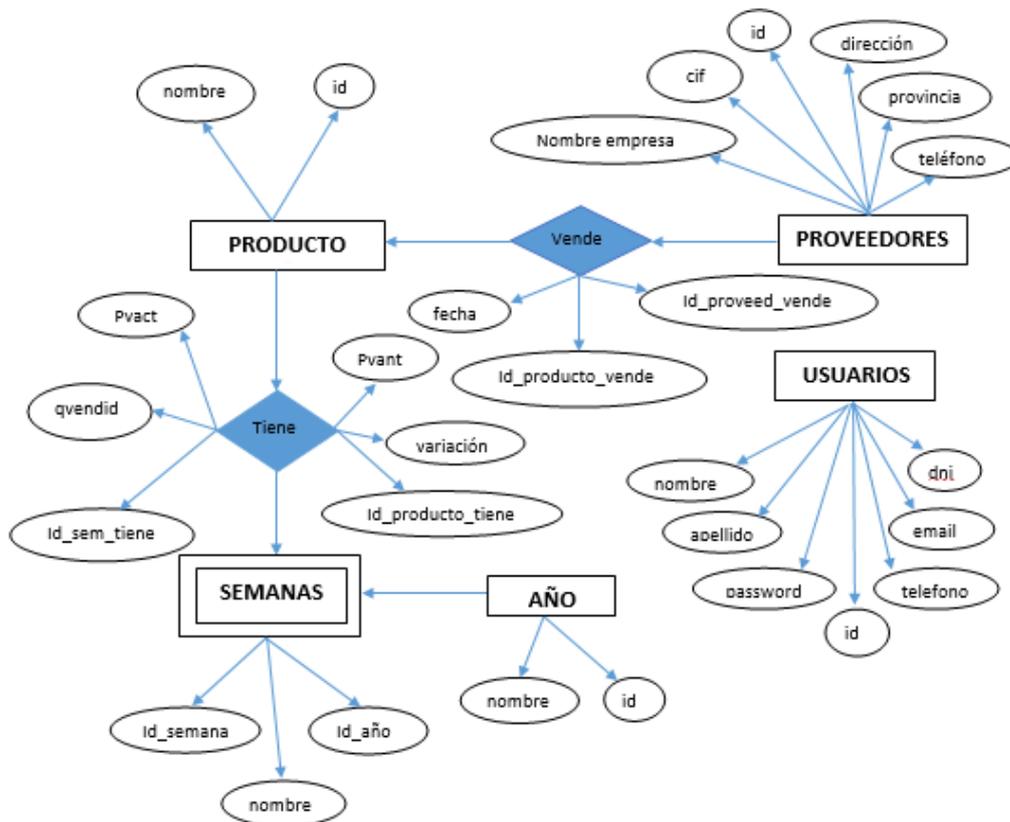


Figura 113. Modelo E-R con atributos, pero sin cardinalidad

Se debe aclarar para un mayor entendimiento que en el modelo E-R, específicamente en la relación “Tiene” posee como atributos “Pvact” que hace referencia al precio de venta de la semana actual, “qvendid” se refiere a la cantidad vendida por semana de los productos que se encuentran a la venta, “Pvant” hace referencia al precio de venta de la semana anterior, “variación” se refiere a la diferencia de precios entre el precio de venta de la semana actual y el precio de venta de la semana anterior.

Por último, se empleará el mismo diagrama E-R, en esta ocasión para representar la cardinalidad de las relaciones, y se expondrá una tabla explicativa con las diferentes razones de por qué se emplea cada cardinalidad [13].

Tabla 1. Cardinalidades

Relación	Descripción
Proveedores VENDEN Productos $(1, n) \rightarrow (1, n)$	Un proveedor puede vender de 1 a varios productos.
	Un producto puede pertenecer de 1 a varios proveedores.
Producto TIENE Semanas $(1, n) \rightarrow (1, n)$	En una semana se van a vender de 1 a varios productos.
	Un producto va a ser vendido de 1 a varias semanas.

El diagrama E-R con las cardinalidades representadas es el siguiente:

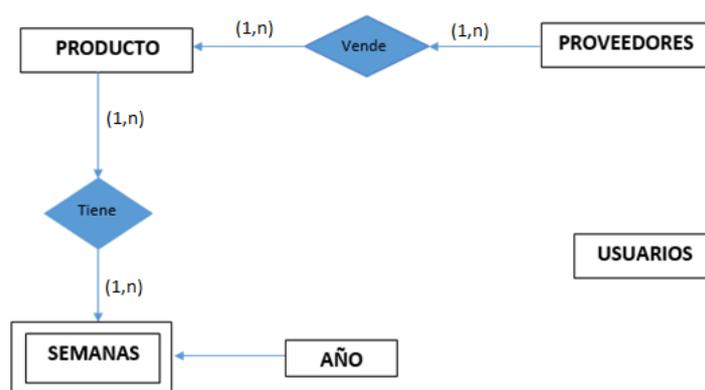


Figura 114. Modelo E-R con cardinalidades

El hecho de que la entidad semanas tenga un doble recuadro lo que indica es que es una entidad débil, y por tanto, queda definida a través de una relación especial que representa la dependencia de esta entidad a otra de orden superior, que en este caso es la entidad año.

A continuación, se explicará cuáles son los atributos de cada entidad, incluidas las dos relaciones que también tienen atributos.

Tabla de atributos

Tabla 2. Proveedores

PROVEEDORES				
Nombre atributo	Alias	Tipo de dato y longitud	V. Defecto	Compuesto
Identificador proveedor	id_proveedor	Int	NOT NULL	NO
Nombre empresa	nomempresa_proveedor	Varchar(40)	NULL	NO
CIF empresa	cif_proveedor	Char(9)	NULL	NO
Dirección empresa	direccion_proveedor	Varchar(100)	NULL	NO
Provincia empresa	provincia_proveedor	Varchar(40)	NULL	NO
Teléfono empresa	telef_proveedor	Int (9)	NULL	NO

Tabla 3. Producto

PRODUCTO				
Nombre atributo	Alias	Tipo de dato y longitud	V. Defecto	Compuesto
Identificador producto	id_producto	Int	NOT NULL	NO
Nombre producto	nom_producto	Varchar(40)	NULL	NO

Tabla 4. Semanas

SEMANAS				
Nombre atributo	Alias	Tipo de dato y longitud	V. Defecto	Compuesto
Identificador semanas	id_semanas	Varchar (4)	NOT NULL	NO
Nombre semanas	nom_semanas	Varchar (40)	NULL	NO
Identificador año	Id_ano	Int	NOT NULL	NO

Tabla 5. Año

AÑO				
Nombre atributo	Alias	Tipo de dato y longitud	V. Defecto	Compuesto
Identificador año	id_ano	Int	NOT NULL	NO
Nombre año	nom_ano	Varchar(40)	NULL	NO

Tabla 6. Usuarios

USUARIOS				
Nombre atributo	Alias	Tipo de dato y longitud	V. Defecto	Compuesto
Identificador usuario	id_usuario	Int	NOT NULL	NO
Nombre usuario	nom_usuario	Varchar(40)	NULL	NO
Apellido usuario	apell_usuario	Varchar(40)	NULL	NO
DNI usuario	dni_usuario	Char(9)	NULL	NO
Email usuario	email_usuario	Varchar(60)	NULL	NO
Teléfono usuario	telef_usuario	Int(9)	NULL	NO
Contraseña usuario	password_usuario	Varchar(60)	NOT NULL	NO

Tabla 7. Vende

VENDE				
Nombre atributo	Alias	Tipo de dato y longitud	V. Defecto	Compuesto
Identificador producto-vende	id_producto_vende	Int	NOT NULL	SI
Identificador proveedor-vende	id_proveedor_vende	Int	NOT NULL	SI
Fecha de venta	Fecha_vende	Formato fecha (date)	NULL	NO

Tabla 8. Tiene

TIENE				
Nombre atributo	Alias	Tipo de dato y longitud	V. Defecto	Compuesto
Identificador producto-tiene	id_producto_tiene	Int	NOT NULL	SI
Identificador semanas-tiene	id_semanas_tiene	Varchar(4)	NOT NULL	SI
Precio de venta anterior	pvant_vende	Double(5,2)	NULL	NO
Precio de venta actual	pvact_vende	Double(5,2)	NULL	NO
Variacion del precio de venta	variac_vende	Double(5,2)	NULL	NO
Cantidad vendida	qvendida_vende	Int	NULL	NO

2. Modelo Lógico de Datos

El Modelo Lógico de Datos (MLD) o Modelo Relacional permite transformar los esquemas que han sido obtenidos en el Modelo Conceptual de Datos en un conjunto de estructuras propias del modelo de datos elegido, en este caso se obtendrá una tabla por relación del MLD cada una de las cuales tendrá un nombre único, independientemente del sistema gestor de bases de datos que se vaya a utilizar.

En cada tabla que va a ser obtenido en el MLD vamos a encontrar los siguientes elementos: atributo, dominio, tupla y relación.

Atributo: Es un elemento susceptible de tomar valores, se corresponde con cada una de las columnas de la tabla.

Dominio: Conjunto de valores que puede tomar un atributo y es considerado finito.

Tupla: Cada uno de los elementos que contiene una instancia de la relación, se corresponde con cada una de las filas de la tabla.

Relación: Corresponde al nombre de la tabla.

En la elaboración del Modelo Lógico de Datos, es necesario seguir el conocido como “Algoritmo de Mapeado”, el cual se compone de 5 pasos que se explicarán a continuación.

Paso 1: Entidades

Una vez definido el término “entidad” (en el apartado 1 del capítulo actual, “Modelo Conceptual de Datos”) para la ejecución del primer paso del algoritmo es necesario crear una relación R, que será una tabla, por cada entidad. Sin embargo, el requisito que exige el algoritmo es que la tabla creada incluya todos los atributos simples, excluyendo a los compuestos, de forma que si hay de este último tipo sólo se indican los simples que lo componen.

Asimismo, se deberá identificar aquellos atributos que funcionan como “primary key” o “clave primaria” de cada entidad, ya que la clave primaria permite identificar de forma única cada registro dentro de una tabla, es por ello que hay que considerar si la entidad es fuerte o débil, ya que si es débil es preciso crear una tabla con sus atributos simples, y estos atributos funcionarán como “foreign key” o “clave externa” de la entidad fuerte asociada. Es por ello que la “primary key” de la entidad considerada débil va a ser la combinación de la clave de la entidad débil y la “foreign key”.

Paso 2: Relaciones

El “Algoritmo de Mapeado” centra su atención en la identificación, cuando corresponda, de las relaciones existentes entre tablas y sus características. Como ya se ha dicho anteriormente, una relación representa una asociación establecida entre campos comunes (columnas) entre dos tablas diferentes.

Existen relaciones de diferentes tipos, que son:

- Relación de uno a varios (1, n): Se crea una relación de este tipo si uno de los campos relacionados es una “clave primaria”. Cada registro de una tabla puede estar enlazado con varios registros de otra, sin embargo, cada registro de esta segunda tabla solo puede estar enlazado con un único registro de la primera tabla. Este tipo de relación es la más común.
- Relación de uno a uno (1, 1): Se crea una relación de uno a uno si ambos campos relacionados son “claves primarias”. En este caso, un registro de una primera tabla sólo puede estar relacionado con otro registro de una segunda tabla y viceversa.
- Relación varios a varios (m, n): En este último caso, ninguno de los campos relacionados es “clave primaria”, por tanto, cada registro de una primera tabla puede estar enlazado con varios registros de una segunda tabla y viceversa. Este tipo de relación implica la repetición de los campos de cada tabla. Para establecer relaciones de este tipo, es necesario la creación de

una tabla intermedia que se encuentre relacionada con las dos de uno a varios.

En el presente proyecto, solo se encuentran relaciones del tipo (1,n) que como ya se ha dicho, es el tipo más común.

Paso 3: Atributos Multivaluados

El “Algoritmo de Mapeado” exige que por cada atributo multivaluado (aquellos atributos que para una misma entidad puede tomar varios valores diferentes, es decir, varios valores del mismo dominio) de una tabla a la que llamaremos “A”, se cree otra tabla “B” referida a ese atributo, de forma que se añada la “primary key” de la tabla “A” como “foreign key” de la tabla “B”.

También es necesario, añadir el atributo multivaluado como monovaluado, por tanto, la “primary key” de la tabla “B” será la combinación de la “foreign key” de la tabla “A” y el atributo multivaluado.

Paso 4: Relación n-arias

De nuevo el “Algoritmo de Mapeado” requiere que por cada relación que asocia a más de dos entidades: se crea una tabla a la que llamaremos de nuevo “A”, se incluyen como atributos las “foreign key” de la tabla “A” (las “claves primarias” de las entidades participantes), la concatenación de todas las “foreign key” serán “claves primarias” de la tabla “A”, y por último, se añaden los atributos de la relación dentro de “A”.

Por ello, si la cardinalidad máxima de una de las entidades es 1, entonces la “clave primaria” de “A” no debe incluir la “clave foránea” de esta entidad.

Paso 5: Especialización o generalización

La generalización es el resultado de la unión de 2 o más conjuntos de entidades de bajo nivel para producir un conjunto de entidades de más alto nivel. La generalización se emplea para resaltar los parecidos entre tipos de entidades de nivel más bajo y ocultar sus diferencias. Para llevar a cabo la generalización es necesario identificar todos aquellos atributos iguales de un conjunto de entidades para formar una o varias entidades globales con dichos atributos semejantes, y por tanto, dicha entidad global quedará a un nivel más alto que las entidades de origen.

La especialización es el resultado de tomar un subconjunto de entidades de alto nivel para formar un conjunto de entidades de más bajo nivel y denota la diferencia entre los conjuntos de entidades de alto y bajo nivel.

La principal diferencia entre especialización y generalización es que en la generalización cada entidad de alto nivel debe ser también una entidad de bajo nivel, mientras que la especialización no tiene este limitante [14].

Por tanto, siguiendo los 5 pasos descritos que componen el “Algoritmo de Mapeado”, se ha construido el “Modelo Lógico de Datos” quedando de la siguiente manera:

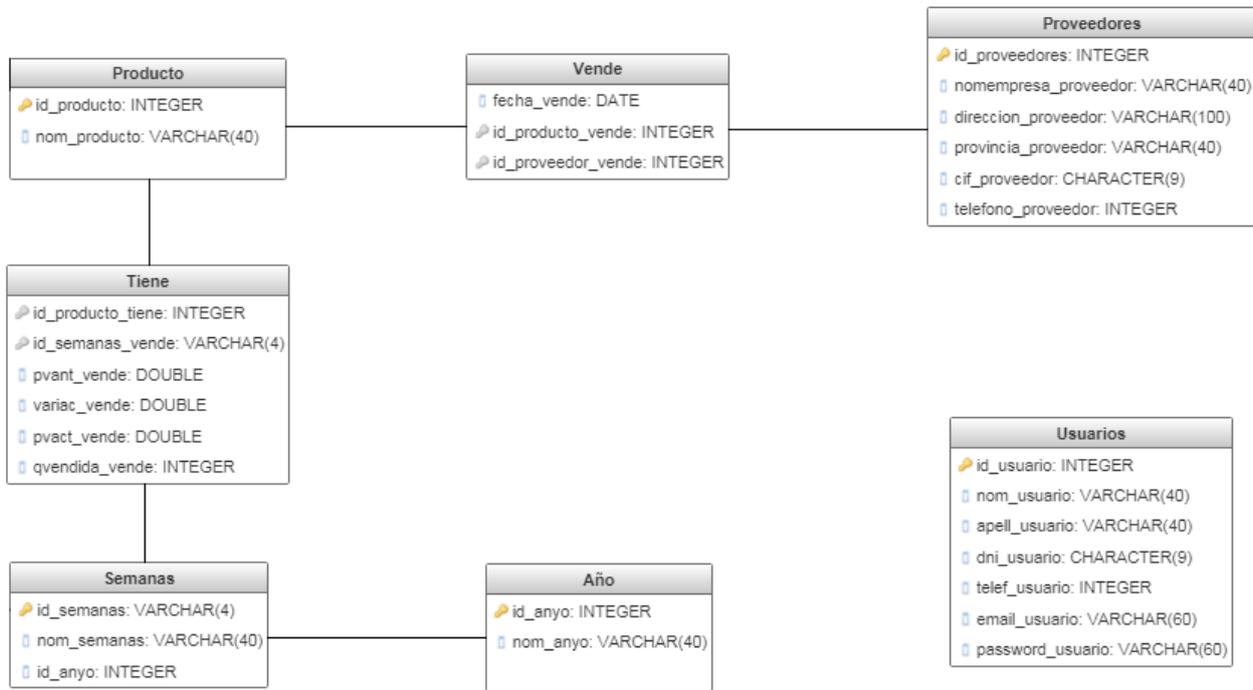


Figura 115. Modelo lógico de datos

A continuación, se indicará cual es la primary key y foreign key, en caso de que la tuviese, de cada tabla de la base de datos.

- La tabla **Proveedores** tiene como primary key “id_proveedor”.
- La tabla **Vende** engloba dos foreign key: “id_producto_vende” y “id_proveedor_vende”.
- La tabla **Producto** posee como primary key “id_producto”.
- La tabla **Tiene** cuenta con dos foreign key: “id_producto_tiene” y “id_semanas_tiene”.
- La tabla **Semanas** incluye como primary key “id_semanas”.
- La tabla **Año** presenta como primary key “id_año”.
- La tabla **Usuarios** dispone como primary key “id_usuario”.

3. Restricciones de Integridad

Como ya se ha dicho el Modelo Relacional es un conjunto finito de relaciones junto con una serie de restricciones o reglas de integridad.

Las restricciones de integridad es la condición necesaria, para preservar la corrección semántica de la base de datos. Se puede encontrar la restricción de integridad asociadas a las tuplas de una relación y asociadas a las relaciones de la base de datos.

1. Restricciones asociadas a las tuplas de la relación:

En ocasiones, se desconoce el valor de un atributo para una determinada tupla (fila de la tabla). Por lo tanto, a dicho atributo se le va a asignar un valor nulo (NULL) que expresa que el valor del mismo es desconocido o que ese atributo no es aplicable a esa tupla.

2. Restricciones asociadas a las relaciones de la base de datos:

En este tipo de restricciones a su vez encontramos varias consideraciones.

2.1 Primary Key: Es el conjunto de atributos seleccionados para identificar unívocamente a las tuplas de una relación. Este tipo de atributos no podrán contener valores nulos (NOT NULL).

2.2 Foreign Key o Clave Externa: Es el conjunto de atributos de una relación cuyos valores en las tuplas deben coincidir con valores de la clave primaria de las tuplas de otra relación. Este tipo de atributos no podrán contener valores nulos (NOT NULL).

2.3 El resto de los atributos podrán contener valores nulos (NULL) debido a que existen casos para cada una de las tablas en los que no se conoce alguno de los datos para los atributos.

2.4 Hay restricciones de dominio en forma de selección única de valor, es decir, solo se puede asignar un valor predefinido.

2.5 Hay restricciones de dominio en forma de selección múltiple de valores, es decir, se puede elegir uno o varios valores predefinidos.

Estas restricciones de integridad proporcionan un medio de asegurar que los cambios que se hacen en la base de datos por usuarios no autorizados no generen una pérdida de la consistencia de los datos [15].

4. Sentencias SQL

Tras la conversión del Modelo Conceptual de Datos al Modelo Lógico de Datos, se procederá a la realización del esquema físico a través una la traducción del MLD para su implementación en un sistema gestor de bases de datos específico. Además, se describirá las relaciones base y estructuras de almacenamiento y métodos de acceso que van a ser empleados para poder acceder a la base de datos de un modo eficiente. Sin embargo, las relaciones base sólo se pueden llevar a cabo cuando el diseñador conozca a la perfección toda la funcionalidad que presenta el sistema gestor de bases de datos que vaya a ser utilizado.

Como ya se ha dicho, el primer paso consiste en realizar la traducción del Modelo Lógico de Datos para que pueda ser implementado por el sistema gestor de bases de datos específico.

Para ello se van a escoger las organizaciones de ficheros más apropiadas para almacenar las relaciones base y los métodos de acceso, a través de un análisis en las transacciones que se van a ejecutar sobre la base de datos.

En el presente proyecto, las sentencias SQL empleadas en las tablas para definir los objetos de la base de datos ha sido “create table”, que como su propio nombre indica permite la creación de una tabla y “default carácter set utf8” que indica la codificación utf8 para los caracteres inmersos en la base de datos.

Para un mayor entendimiento de las tablas, se explicará los tipos de datos que han sido empleados, así como la diferencia entre las restricciones NULL y NOT NULL.

Respecto a los tipos de datos empleados han sido los siguientes:

Cadena VARCHAR: Las cadenas de texto son de tipo varchar, y su tamaño máximo para un determinado campo se especifica indicando su longitud entre paréntesis. Por ejemplo, en el diseño de la tabla “Usuarios”, es necesario valorar que la longitud máxima se va a establecer para la “contraseña” pues se ha decidido que como sumo la contraseña no sobrepase los 60 letras o caracteres, de forma que el tipo de dato para este campo se definió como VARCHAR (60) y por tanto, el campo “contraseña” puede almacenar cualquier cadena alfanumérica de hasta 60 caracteres.

Número entero INT: A diferencia del tipo VARCHAR, donde se establece el tamaño máximo, para los números enteros existen varios tipos de dato de tamaño fijo, de forma que se elegirá uno u otro en función del tamaño máximo que necesitamos establecer. Es por ello que cuando se habla de números, lo más correcto es hablar de rango en lugar de tamaño, es decir, bajo que rango de valores (máximo y mínimo) se puede operar con este tipo.

Numero decimal DOUBLE: Para los números decimales también existen varios tipos de dato con diferente rango de valores posibles, la parte entera se separa por un punto de la decimal. Es muy similar al tipo de dato FLOAT diferenciándose únicamente en que el tipo FLOAT presenta un tamaño en bytes de 4 mientras que el tipo DOUBLE tiene un tamaño de 8 bytes.

Fecha DATE: El tipo DATE tiene el tamaño apropiado para registrar un dato de año+mes+día. Aunque no ha sido empleado también existe otro tipo de dato denominado DATETIME que además del día también registra la hora.

En cuanto a la diferencia entre las restricciones NULL y NOT NULL, es simple, la restricción NOT NULL sirve para especificar que una columna no acepta el valor NULL (nulo), es decir, que esa columna siempre va a tener que tener algún valor, no puede estar vacía. Y por consiguiente la restricción NULL, implica que la columna no tiene que tener ningún valor y, por tanto, estará vacía.

Las claves primarias deben ser únicas y no nulas, de forma que garanticen que una fila de una tabla pueda ser siempre referenciada a través de su clave primaria, y por ello, es necesario que se especifique como NOT NULL aquellas columnas que se van a utilizar como claves primarias cuando se va a crear una tabla.

Capítulo VIII

Gráfico D3.js

D3.js es la librería de JavaScript que permite crear visualizaciones complejas y gráficos interactivos. Esta librería permite manipular documentos basados en datos empleando estándares abiertos de la web y los navegadores pueden crear visualizaciones complejas sin depender de un software propietario. Presenta la gran ventaja de que los desarrollos se encuentran abiertos y por tanto, pueden ser reimplementados por otros desarrolladores.

1. Selección del gráfico

El proceso de selección del tipo de gráfico para su representación con D3.js fue algo complejo debido a la extensa variedad de gráficos que permiten la Visualización de Datos.

El proceso de selección comenzó con un análisis de los requisitos de la empresa. Una vez que la empresa comunicó sus necesidades y el requerimiento que debía tener el gráfico, se escogió aquellos gráficos de D3.js que cumplían dichos requisitos.

Como ya se ha dicho anteriormente, “Mafrilasa” pertenece al sector secundario de la industria ligera de alimentación. En este tipo de industrias los precios no permanecen constantes, sino que presentan continuas variaciones por semanas. Atendiendo a esta justificación, el requisito principal que se exigió en el desarrollo del proyecto fue el de poder visualizar los distintos precios de cada uno de los productos, tanto el precio de la semana actual como el de las semanas anteriores.

Por tanto, analizando los diferentes gráficos que ofrece D3.js como herramienta para la Visualización de Datos, se llegó a que los que mejor se adaptaban a los requisitos eran: Calendar View, Line Chart o Bar Chart.

La opción 1, Calendar View, se evaluó como posibilidad por el hecho de que en este gráfico los valores de los precios se visualizarían como celdas de colores por día. Los días se organizan en columnas por semana, luego se agrupan por mes y año. De forma que cuando el usuario quisiese conocer el precio de una semana situaría el puntero sobre la semana del año del siguiente modo:

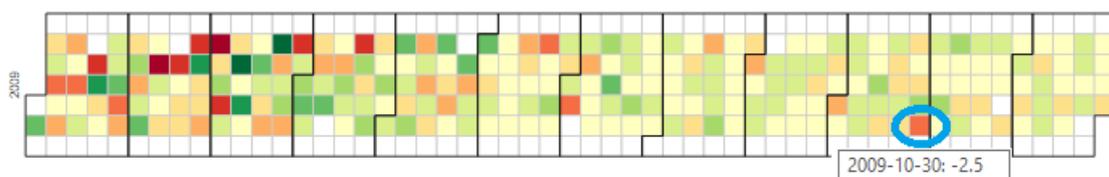


Figura 116. Análisis Calendar View

Sin embargo, el motivo por el que no se eligió es porque a pesar de que Calendar View es buena opción, no ofrecía al usuario la posibilidad de ver la variación que sufren los precios respecto a la semana anterior.

La opción 2, Bar Chart, se evaluó como posibilidad por el hecho de que este tipo de gráfico permitiría representar el precio respecto a las 52 semanas del año, y sí que se tendría la variación respecto a la semana anterior que era uno de los objetivos, sin embargo, se pensó en que sería mejor poner en el máximo de cada barra un punto y luego unir los diferentes puntos, para que fuese más visual para el usuario la variación que sufren los precios. La idea era la siguiente:

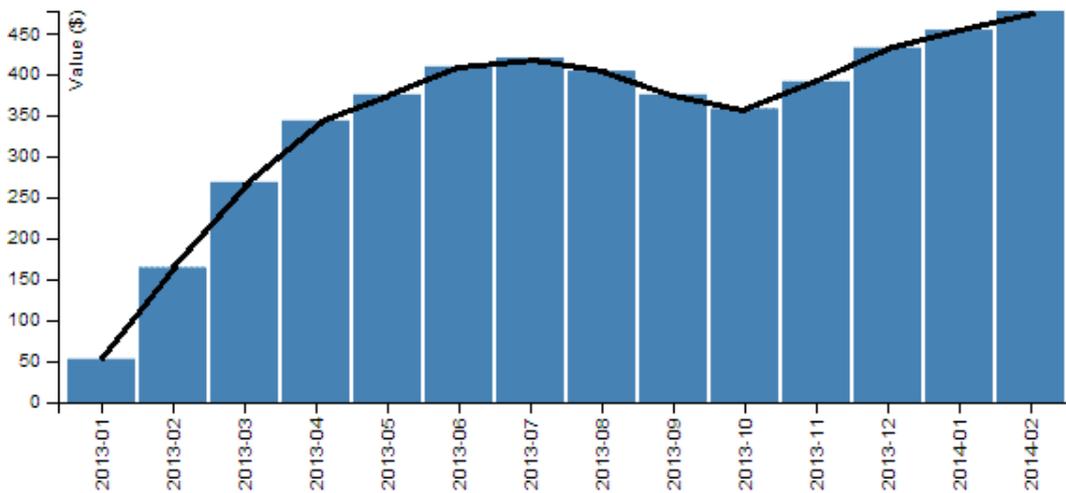


Figura 117. Análisis Bar Chart

No obstante, si se añadía una línea que uniese los máximos de cada barra del gráfico Bar Chart para tener una visión global de los movimientos de los precios en el mercado, se estaría realizando un gráfico del tipo Line Chart, por lo que este fue el gráfico seleccionado.

Un Gráfico de Líneas o Line Chart es una conexión entre varios “marcadores” de puntos de datos por segmentos de línea. Consta del eje X que en este caso representa las 52 semanas del año, y el eje Y que muestra los precios en €/kg.

La ventaja principal de los Line Chart, es que permiten observar rápida y cómodamente cual es la tendencia en el mercado durante un periodo determinado de tiempo. Otra de las ventajas se produce al posicionar el puntero sobre cada uno de los puntos que forman el gráfico, puesto que se muestra al usuario el precio exacto de esa semana del siguiente modo:

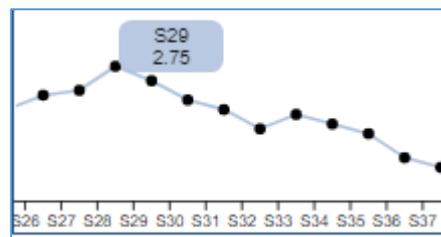


Figura 118. Muestra de precio y semana en cada punto del Line Chart

2. Pantalla Menu_grafica.php

La pantalla Menu_grafica.php muestra un menú desplegable de los diferentes productos que tiene “Mafrilasa” a la venta. De forma, que cuando el usuario quiera acceder a la visualización de alguna de las gráficas con el menú, podrá seleccionar aquel producto que desee y pulsando sobre “Estadísticas” accederá a la gráfica.



Figura 119. Menu_grafica.php

3. Análisis Pantalla Menu_grafica.php

El código comienza con dos cadenas de caracteres que en PHP que muestran el estilo que tendrá la página con PHP, se representa a través de “echo”.

Para crear la conexión con el servidor de la base de datos se emplea la función `mysqli_connect()` y los datos o variables que se acaban de crear. Para ello se crea una variable `$conexion` que contendrá la conexión de forma que queda:

```
$conexion = mysqli_connect($host_db, $user_db, $pass_db, $db_name);
```

A continuación, le sigue el condicional que devuelve TRUE en caso de conexión satisfactoria con MariaDB o FALSE en caso de error.

```
if (!$conexion) {
    die("La conexión falló: " . mysqli_error());
}
else
{
    echo "Conexión satisfactoria";
}
```

Le sigue varias cadenas de caracteres a través de cuatro sentencias `echo`, de las cuales la primera indica el estilo, la segunda realiza una llamada a la página `datos_mysql.php` a través de `<form action="datos_mysql.php" method="POST">`, la tercera establece el estilo de que debe tener la etiqueta y la cuarta establece el estilo que debe tener cada producto del desplegable creado.

A continuación, un bucle `while` que permitirá mostrar todos los productos en el desplegable creado.

```

while($row = mysqli_fetch_row($result)){
    echo "<option value='". $row[0]."'>". $row[1]."</option>\n";
}

```

Finalmente, para cerrar la conexión de la base se utiliza la función `mysqli_close()`.

```

<?php
echo "<div style='font-family: Arial, sans-serif; text-align: center; font-size: 50px;
font-weight: bold; color: #075166; text-shadow: 0 8px 9px #c4b59d, 0px -2px 1px #fff; border-radius: 20px; height: 10%;>Gráficas</div>";
echo "<div style='background-color: #075166; height: 7%; width: 100%;'> </div>";

$host_db = "localhost";
$user_db = "root";
$pass_db = "";
$db_name = "tfg_db";
$tbl_name = "producto";

$conexion = mysqli_connect($host_db, $user_db, $pass_db, $db_name);
$conexion->set_charset("utf8");

if (!$conexion) {
    die("La conexión falló: " . mysqli_error());
}

$sql = "SELECT id_producto, nom_producto FROM $tbl_name order by id_producto";
$result = mysqli_query($conexion, $sql);

echo "<div style='background-color:#0489B1; border-radius: 8px; box-shadow: 3px 3px 10px 0px rgba(50, 50, 50, 0.75); -moz-box-shadow:
3px 3px 10px 0px rgba(50, 50, 50, 0.75); -webkit-box-shadow: 3px 3px 10px 0px rgba(50, 50, 50, 0.75); margin-left: auto; margin-right:
auto; margin-top: 5%; margin-bottom: 5%; max-width: 20em; padding-bottom: 10px; padding-top: 10px;'>";
echo "<form action='datos_mysql.php' method='POST'>";
echo "<label style='font-family: calibri, sans-serif; color: white; display: block; margin-bottom: 6px;margin-left: 40px; '>Producto: </label>";
echo "<select name='id_prod' style='margin-bottom: 20px; margin-left: 40px; width:70%'>";
while($row = mysqli_fetch_row($result)){
    echo "<option value='". $row[0]."'>". $row[1]."</option>\n";
}
echo "</select>";
echo "<input type='submit' style='text-decoration: none; padding: 10px; font-weight: 600; font-size: 15px; color: #ffffff; background-color:
#075166; border-radius: 6px; width: 50%; text-align: center; margin-top:5%; margin: 0 auto; display:block;' value='Estadísticas'>";
echo "</form></div>";

mysqli_close($conexion);
echo "<a style='font-family: calibri, sans-serif; margin-bottom: 10%; text-decoration: none; padding: 10px; font-weight: 600;
font-size: 20px; color: #ffffff; background-color: #1883ba; border-radius: 6px; margin: 0 auto; display: block; width: 15%; text-align:
center; margin-top:5%;' href='menu.html'>Atrás</a>";
?>

```

Figura 120. Código `Menu_grafica.php`

4. Pantalla `Graficas.html`

La pantalla `Graficas.html` muestra como resultado la evolución de los precios a lo largo de las 52 semanas que componen un año, de un determinado producto que haya sido seleccionado por el usuario del menú desplegable de la pantalla “Menú Gráficas”

Asimismo, bajo la gráfica se encontrará una tabla que contiene datos como la semana, el precio antiguo que corresponde al precio que tuvo ese producto la semana anterior, el precio actual y la cantidad en kilogramos que se ha vendido esa semana. Estos cinco parámetros mencionados permiten facilitar la comprensión del gráfico usuario.

A continuación, se mostrarán tres ejemplos correspondientes a tres productos diferentes:

Producto 1: Costillejas



Figura 121. Gráfica Costillejas - Ejemplo Producto 1

Producto 2: Morro

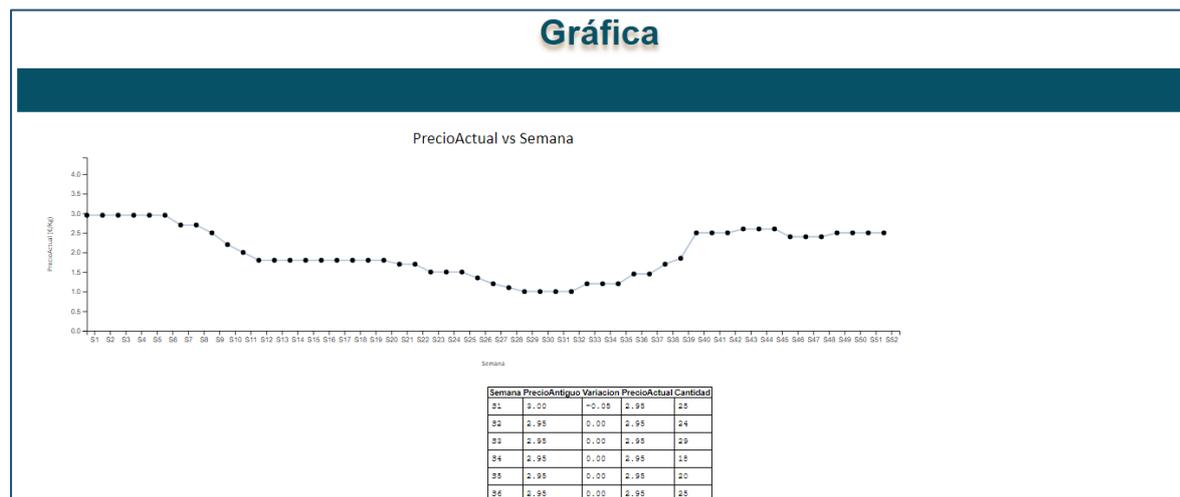


Figura 122. Gráfica Morro - Ejemplo Producto 2

Producto 3: Jamón 3D S/H

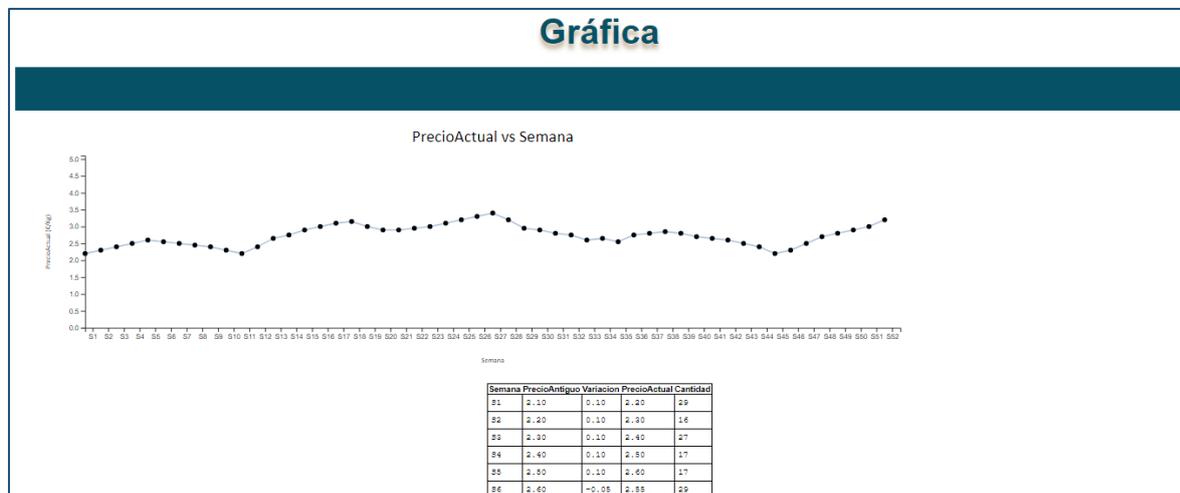


Figura 123. Gráfica Jamón 3D S/H - Ejemplo Producto 3

5. Análisis Pantalla Gráficas.html

La primera parte del código HTML lo forma el <head>no se explicará ya que es exactamente igual, que los códigos explicados en el capítulo VI y a mayores, contiene el estilo de cada uno de los elementos que forman el gráfico y la tabla de cada producto.

El código D3.js comienza con el paso 1, que consiste en establecer las dimensiones y los márgenes del gráfico D3.js, de modo que la primera línea define los cuatro márgenes que rodean el gráfico.

```
var margin = {top: 40, right: 100, bottom: 80, left: 100},
    width = screen.width - margin.left - margin.right,
    height = screen.height - 400 - margin.top - margin.bottom;
```

En el paso 2, primero se define la línea que se dibuja empleando la función `d3.line()`, en segundo lugar se establecen los tipos de escalas para los ejes, de manera que para el eje x se quiere una escala por etiquetas, ya que se requiere que en la gráfica de la x aparezcan las 52 semanas, de la siguiente forma S1,S2,S3... En cuanto al eje y, se requiere una una escala numérica. Es por ello, que para el eje "x" se pondrá `d3.scaleBand()`; y para el eje "y", `var y = d3.scaleLinear().range([height, 0]);`.

El paso 3 consiste en definir el div para el tooltip, es decir, se define el div que va a contener los datos de la semana y el precio, al posicionar el cursor sobre alguno de los puntos que forman la gráfica, del siguiente modo:

```
var divTooltip = d3.select("body").append("divTooltip")
    .attr("class", "tooltip")
    .style("opacity", 0);
```

En el paso 4, se anexionan los atributos SVG y elementos del grupo empleando el siguiente código:

```
var svg = d3.select("body")
    .append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
    .append("g")
    .attr("transform", "translate(" + margin.left + "," + margin.top + ")");
```

El paso 5, consiste en leer los datos, es decir, ya se puede leer datos del conjunto de datos data.csv.

```
d3.csv("data.csv", function(error, data) {
    if (error) throw error;
```

El paso 6 se basa en formatear los datos. El código utilizado asegura que todos los valores que se extraen del archivo csv se configuran y formatean correctamente.

```
data.forEach(function(d) {
    d.Semana=d.semana;
    d.PrecioActual = d.PrecioActual;
    d.variaciones = d.variaciones;
    d.cantidades = d.cantidades;
    d.PrecioActualAntiguos = d.PrecioActualAntiguos;
});
```

En el paso 7, se establece el rango de la escala. Después de formatear los datos, se define el rango de escala para “x” e “y”.

```
x.domain(data.map(function(d) { return d.Semana; })).range([0, width],
10.5);;
y.domain([0, d3.max(data, function(d) { return d.PrecioActual;
})*1.5]);
```

El paso 8, consiste en anexar la ruta y datos, del siguiente modo:

```
svg.append("path")
    .data([data])
    .attr("class", "line")
    .attr("d", valueline);

svg.append("text")
    .attr("x", (width / 2))
    .attr("y", 0 - (margin.top / 2))
    .attr("text-anchor", "middle")
    .style("font-size", "24px")
    .text("PrecioActual vs Semana");
```

En el paso 9, consiste en asociar el tooltip que ha sido creado en el paso 3 a los puntos de la gráfica.

```
svg.selectAll("dot")
    .data(data)
    .enter().append("circle")
    .attr("r", 3.5)
    .attr("cx", function(d) { return x(d.Semana); })
    .attr("cy", function(d) { return y(d.PrecioActual); })
    .on("mouseover", function(d) {
        divTooltip.transition()
            .duration(200)
            .style("opacity", .9);
        divTooltip.html(d.Semana+"<br/>" + d.PrecioActual)
```

```

        .style("left", (d3.event.pageX) + "px")
        .style("top", (d3.event.pageY - 28) + "px")
    })
    .on("mouseout", function(d) {
        divTooltip.transition()
            .duration(500)
            .style("opacity", 0);
    });

```

El paso 10, se fundamenta en agregar el eje “x” y la etiqueta que acompañará al eje, que es este caso será “Semana” y en agregar el eje “y” etiqueta que acompaña al eje, que será “Precio Actual (€/kg)”.

Eje “x”:

```

    svg.append("g")
        .attr("transform", "translate(0," + height + ")")
        .attr("class", "axis")
        .call(d3.axisBottom(x));
    svg.append("text")
        .attr("transform",
            "translate(" + (width/2) + "," +
            (height + margin.top + 10) + ")")
        .style("text-anchor", "middle")
        .text("Semana");

```

Eje “y”:

```

    svg.append("g")
        .attr("class", "axis")
        .call(d3.axisLeft(y));
    svg.append("text")
        .attr("transform", "rotate(-90)")
        .attr("y", 40 - margin.left)
        .attr("x", 0 - (height / 2))
        .attr("dy", "1em")
        .style("text-anchor", "middle")
        .text("PrecioActual (€/Kg)");

```

Con esta secuencia de pasos, se obtendrán las diferentes gráficas para cada producto que se pueden visualizar en Graficas.html.

Como ya se ha dicho anteriormente, justo debajo de cada tabla el usuario encontrará una tabla que facilite la comprensión del gráfico.

Las tablas creadas, también han sido implementadas con D3.js, por ello se explicará el código utilizado que es continuación del anterior.

El primer paso consiste en el empleo de la función “*tabulate*” que permite la creación de tablas con D3.js.

```
function tabulate(data, columns) {
    var table = d3.select("body").append("table"),
        thead = table.append("thead"),
        tbody = table.append("tbody");
```

En el paso 2, se añade la fila encabezado de la tabla, siguiendo el código:

```
thead.append("tr")
    .selectAll("th")
    .data(columns)
    .enter()
    .append("th")
    .text(function(column) { return column; });
```

En el paso 3, se crea una fila para cada objeto en los datos, del siguiente modo:

```
var rows = tbody.selectAll("tr")
    .data(data)
    .enter()
    .append("tr");
```

En el paso 4, se origina cada una de las celdas de las que va a constar la tabla, de forma que se genera una celda en cada fila para cada columna.

```
var cells = rows.selectAll("td")
    .data(function(row) {
        return columns.map(function(column) {
            return {column: column, value: row[column]};
        });
    })
    .enter()
    .append("td")
    .attr("style", "font-family: Courier") // sets the font style
    .html(function(d) { return d.value; });
```

El último de los pasos consiste en establecer el nombre para cada una de las columnas de las que va a constar la tabla.

```
tabulate(data, ["Semana", "PrecioAntiguo", "Variacion", "PrecioActual", "Cantidad"]);
```

El código completo se puede ver en las siguientes figuras, ya que debido a que su longitud en una única figura no es posible verlo.

```

<html>
  <head>
    <title>Gráfica</title>
    <link rel="stylesheet" type="text/css" href="estilo.css">
    <meta http-equiv="Content-Type" content="text/html"; charset="utf-8"/>
    <style /* set the CSS */
      body { font: 10px calibri;}

      .line {
        fill: none;
        stroke: lightsteelblue;
        stroke-width: 2px;
      }
      divTooltip.tooltip {
        position: absolute;
        text-align: center;
        width: 60px;
        height: 28px;
        padding: 2px;
        font: 12px sans-serif;
        background: lightsteelblue;
        border: 0px;
        border-radius: 8px;
        pointer-events: none;
      }
      table {
        border-collapse: collapse;
        border: 2px black solid;
        font: 12px sans-serif;
        margin: 0 auto;
      }
      td {
        border: 1px black solid;
        padding: 5px;
      }
    </style>
  </head>
  <body>
    <div class="titulo">Gráfica</div>
    <div id="menu_inicio"> - </div>

    <br/><br/>

    <!-- load the d3.js library -->
    <script src="https://d3js.org/d3.v4.min.js"></script>
    <script>
      // set the dimensions and margins of the graph
      var margin = {top: 40, right: 100, bottom: 80, left: 100},
          width = screen.width - margin.left - margin.right,
          height = screen.height - 400 - margin.top - margin.bottom;

      // set the ranges
      var x = d3.scaleBand();
      var y = d3.scaleLinear().range([height, 0]);

      // define the line
      var valueline = d3.line()
        .x(function(d) { return x(d.Semana); })
        .y(function(d) { return y(d.PrecioActual); });

      // Define the div for the tooltip
      var divTooltip = d3.select("body").append("divTooltip")
        .attr("class", "tooltip")
        .style("opacity", 0);
    </script>
  </body>
</html>

```

Figura 124. Código Gráfico.html Parte 1

```

</style>
</head>
<body>

  <div class="titulo">Gráfica</div>
  <div id="menu_inicio"> - </div>

  <br/><br/>

  <!-- load the d3.js library -->
  <script src="https://d3js.org/d3.v4.min.js"></script>
  <script>
    // set the dimensions and margins of the graph
    var margin = {top: 40, right: 100, bottom: 80, left: 100},
        width = screen.width - margin.left - margin.right,
        height = screen.height - 400 - margin.top - margin.bottom;

    // set the ranges
    var x = d3.scaleBand();
    var y = d3.scaleLinear().range([height, 0]);

    // define the line
    var valueline = d3.line()
      .x(function(d) { return x(d.Semana); })
      .y(function(d) { return y(d.PrecioActual); });

    // Define the div for the tooltip
    var divTooltip = d3.select("body").append("divTooltip")
      .attr("class", "tooltip")
      .style("opacity", 0);
  </script>
</body>
</html>

```

Figura 125. Código Gráfico.html Parte 2

```

// append the svg object to the body of the page
// appends a 'group' element to 'svg'
// moves the 'group' element to the top left margin
var svg = d3.select("body")
    .append("svg")
      .attr("width", width + margin.left + margin.right)
      .attr("height", height + margin.top + margin.bottom)
    .append("g")
      .attr("transform",
        "translate(" + margin.left + "," + margin.top + ")");

// Get the data
d3.csv("data.csv", function(error, data) {
  if (error) throw error;

  // format the data
  data.forEach(function(d) {
    d.Semana = d.Semana;
    d.PrecioActual = d.PrecioActual;
    d.variaciones = d.variaciones;
    d.cantidades = d.cantidades;
    d.PrecioActualAntiguos = d.PrecioActualAntiguos;
  });

  // Scale the range of the data
  x.domain(data.map(function(d) { return d.Semana; })).range([0, width], 10.5);
  y.domain([0, d3.max(data, function(d) { return d.PrecioActual; })*1.5]);

  // Add the valueline path.
  svg.append("path")
    .data([data])
    .attr("class", "line")
    .attr("d", valueline);

```

Figura 127. Código Gráfico.html Parte 3

```

svg.append("text")
  .attr("x", (width / 2))
  .attr("y", 0 - (margin.top / 2))
  .attr("text-anchor", "middle")
  .style("font-size", "24px")
  .text("PrecioActual vs Semana");

// Add the scatterplot
svg.selectAll(".dot")
  .data(data)
  .enter().append("circle")
  .attr("r", 3.5)
  .attr("cx", function(d) { return x(d.Semana); })
  .attr("cy", function(d) { return y(d.PrecioActual); })
  .on("mouseover", function(d) {
    divTooltip.transition()
      .duration(200)
      .style("opacity", .9);
    divTooltip.html(d.Semana + "<br/>" + d.PrecioActual)
      .style("left", (d3.event.pageX) + "px")
      .style("top", (d3.event.pageY - 28) + "px")
  })
  .on("mouseout", function(d) {
    divTooltip.transition()
      .duration(500)
      .style("opacity", 0);
  });

```

Figura 126. Código Gráfico.html Parte 4

```

// Add the X Axis
svg.append("g")
  .attr("transform", "translate(0," + height + ")")
  .attr("class", "axis")
  .call(d3.axisBottom(x));

// text label for the X axis
svg.append("text")
  .attr("transform",
    "translate(" + (width/2) + " ," +
    (height + margin.top + 10) + ")")
  .style("text-anchor", "middle")
  .text("Semana");

// Add the Y Axis
svg.append("g")
  .attr("class", "axis")
  .call(d3.axisLeft(y));

// text label for the y axis
svg.append("text")
  .attr("transform", "rotate(-90)")
  .attr("y", 40 - margin.left)
  .attr("x", 0 - (height / 2))
  .attr("dy", "1em")
  .style("text-anchor", "middle")
  .text("PrecioActual (€/Kg)");

```

Figura 129. Código Gráfico.html Parte 5

```

// The table generation function
function tabulate(data, columns) {
  var table = d3.select("body").append("table"),
      thead = table.append("thead"),
      tbody = table.append("tbody");

  // append the header row
  thead.append("tr")
    .selectAll("th")
    .data(columns)
    .enter()
    .append("th")
    .text(function(column) { return column; });

  // create a row for each object in the data
  var rows = tbody.selectAll("tr")
    .data(data)
    .enter()
    .append("tr");

  // create a cell in each row for each column
  var cells = rows.selectAll("td")
    .data(function(row) {
      return columns.map(function(column) {
        return {column: column, value: row[column]};
      });
    })
    .enter()
    .append("td")
    .text(function(d) { return d.value; });
}

```

Figura 128. Código Gráfico.html Parte 6

```
.enter()
.append("td")
.attr("style", "font-family: Courier") // sets the font style
.html(function(d) { return d.value; });

return table;
}

// render the table
tabulate(data, ["Semana", "PrecioAntiguo", "Variacion", "PrecioActual", "Cantidad"]);
});
</script>

</body>
```

Figura 130. Código Gráfico.html Parte 7

Capítulo IX

Manual de usuario

El presente manual de usuario tiene como principal objetivo dar a conocer de un modo detallado y sencillo la estructura de la página web creada para la PYME “Mafrilasa”, de forma que el usuario pueda aprovechar al máximo el entorno web diseñado.

Página Principal

Para comenzar el usuario debe ingresar la dirección electrónica <http://localhost:8080/TFG/principal.html> para visualizar la página web de “Mafrilasa”.

Una vez que el usuario ingresa en el sitio web, lo primero que se encontrará es la página principal de la web, donde el usuario deberá introducir su nombre de usuario y contraseña para acceder al menú.



The screenshot shows a web page with a dark teal header containing the word "EMPRESA" in white. Below the header is a white login form with a dark teal border. The form contains two input fields: "Usuario:" and "Contraseña:". Below these fields is a dark teal button labeled "Enviar".

Figura 131. Manual de usuario Página Principal

Menú Principal

Una vez introducidos el usuario y la contraseña, se accederá al menú principal, teniendo la posibilidad de ingresar a las diferentes opciones disponibles de la web.



The screenshot shows a web page with a dark teal header containing the word "Menú" in white. Below the header is a white menu form with a dark teal border. The form contains three input fields: "Gestión de proveedores", "Gestión de productos", and "Gráficas". Below these fields is a dark teal button labeled "Atrás".

Figura 132. Manual de usuario Menú Principal

Menú de Productos

Si por el contrario el usuario selecciona “Gestión de productos” se encontrará con un submenú, en el que podrá seleccionar de nuevo 6 opciones diferentes tal y como se ve en la figura 133.

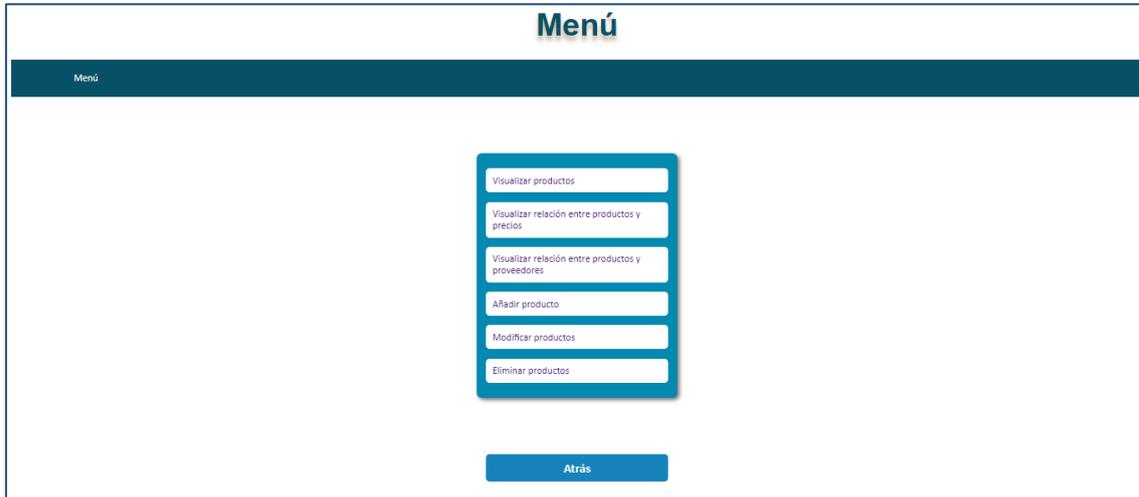


Figura 133. Manual de usuario Menú Productos

Opción Visualizar Productos

Escogiendo “Visualizar Productos” del submenú “Gestión de Productos”, el usuario tendrá la posibilidad de visualizar cuales son los productos que pone a la venta la empresa, así como el código de cada producto.

Gráficas	
Código	Nombre del producto
62	SALCHICHA de POLLO 2 kilos
96	MAGRO Limpieza de chuletero 7525 min 25
2037	SOLOMILLOS(VACIO) 5 unidades
2055	CARRILLERA DE CERDO VACIO 1KG
2115	CONTRA extratierna 2 trozos vacio
2124	LOMO EXTRATIERNO (VACIO)
2126	SOLOMILLO PR. EXTRATIERNO (VACIO)
2131	CHULET sin cabec sin solomillo Extratier
2132	LOMO ADOBADO EXTRATIERNO
2142	JAMON ADOBADO EXTRATIERNO Deshues

Figura 134. Manual de usuario Opción Visualizar Productos

Opción Visualizar relación entre Productos y Proveedores

Escogiendo “Visualizar relación entre Productos y Proveedores” del submenú “Gestión de Productos”, el usuario accederá a la misma pantalla “Visualizar relación entre Productos y Proveedores” del submenú “Gestión de Proveedores”. Por tanto, esta pantalla se podrá ver más adelante.

Opción Visualizar relación entre Productos y Precios

Escogiendo “Visualizar relación entre Productos y Precios” del submenú “Gestión de Productos”, el usuario tendrá la posibilidad de visualizar el nombre del producto, así como su precio anterior, su precio actual, la variación de precio y la cantidad en kilogramos, a lo largo de las 52 semanas que componen el año.

Relación productos y precios					
Producto	Precio anterior	Variación	Precio actual	Semanas	Cantidad (kg)
SALCHICHA de POLLO 2 kilos	4.05	0.05	4.10	S1	28
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S10	28
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S11	27
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S12	20
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S13	26
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S14	20
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S15	20
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S16	22
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S17	22
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S18	20
SALCHICHA de POLLO 2 kilos	4.10	0.05	4.15	S19	29
SALCHICHA de POLLO 2 kilos	4.10	0.00	4.10	S2	25

Figura 135. Manual de usuario Opción Visualizar relación entre productos y precios

Opción Añadir Producto

Escogiendo “Añadir Producto” del submenú “Gestión de Productos”, el usuario tendrá la posibilidad de añadir un nuevo producto a la base de datos de la empresa, introduciendo únicamente el nombre del nuevo producto y su código.

PRODUCTO

Menú

Código producto:

Nombre producto:

Figura 136. Manual de usuario Opción Añadir Producto

Opción Eliminar Producto

Escogiendo “Eliminar Producto” del submenú “Gestión de Productos”, el usuario tendrá la posibilidad de eliminar un producto a la base de datos de la empresa, introduciendo únicamente el código del producto que se desea eliminar, tal y como se puede ver en la figura 137.



Figura 137. Manual de usuario Opción Eliminar Producto

Opción Modificar Producto

Escogiendo “Modificar Producto” del submenú “Gestión de Productos”, el usuario tendrá la posibilidad de modificar un producto de la base de datos de la empresa. Aquello que se desee que se mantenga igual dentro de los datos, se dejará vacío de manera que solo se introducirán los cambios nuevos. Lo que es imprescindible de introducir es el código del proveedor.



Figura 138. Manual de usuario Opción Modificar Producto

Menú de Proveedores

Si el usuario decide seleccionar “Gestión de proveedores”, se encontrará con un submenú, en el que podrá seleccionar 5 opciones diferentes tal y como se ve en la figura 139.

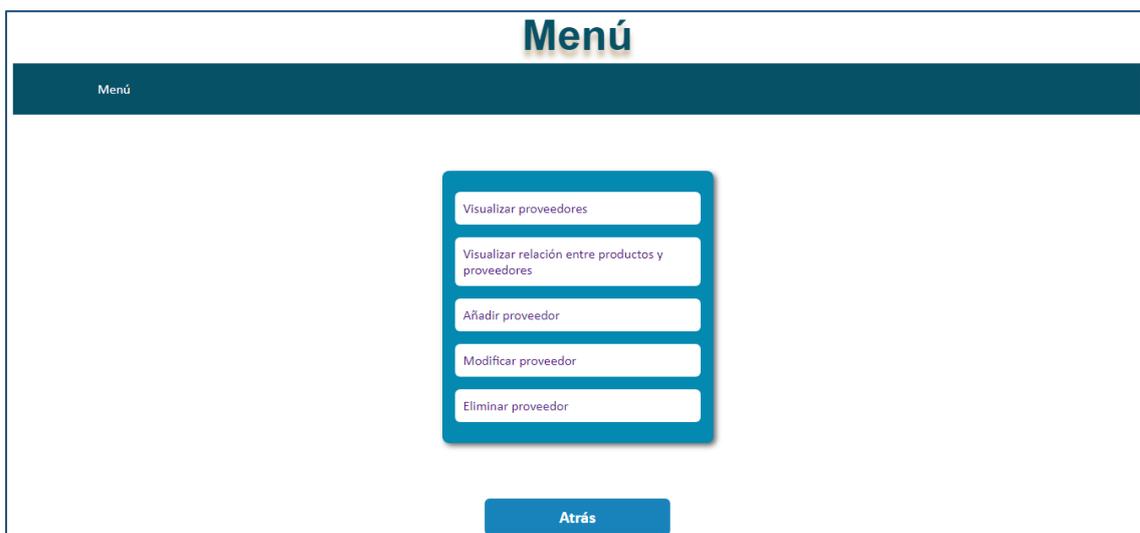


Figura 139. Manual de usuario Menú Proveedores

Opción Visualizar Proveedores

Escogiendo “Visualizar Proveedores” del submenú “Gestión de Proveedores”, el usuario tendrá la posibilidad de visualizar los datos correspondientes a los diferentes proveedores con los que cuenta la empresa. Entre los datos se encuentran el código, nombre, dirección, provincia, CIF y teléfono correspondiente a cada uno de los proveedores.

Código	Nombre	Dirección	Provincia	CIF	Teléfono
1	Navarro Montes	calle tratado de tordesillas n98	valladolid	71171119x	674329064
2	Hermanos Gil	calle bartolome de las casas n21	valladolid	71648564x	645709327
3	Maya Blue	calle valentín calderon n17	valladolid	71753495x	692875934
4	García Rey	calle primera universidad n43	valladolid	71457932x	601294390

[Atrás](#)

Figura 140. Manual de usuario opción Visualizar Proveedores

Opción Visualizar relación entre Productos y Proveedores

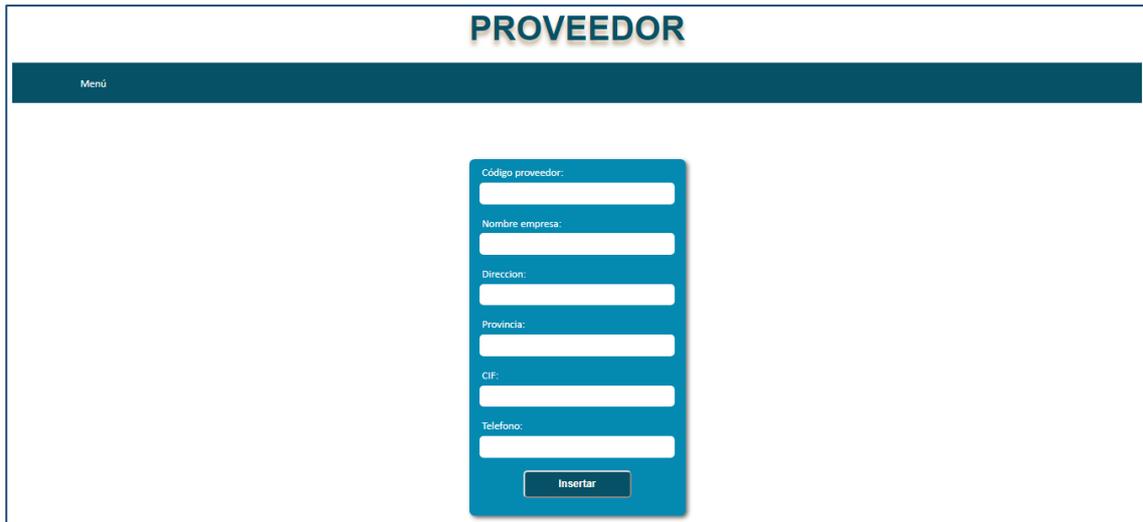
Escogiendo “Visualizar relación entre Productos y Proveedores” del submenú “Gestión de Proveedores”, el usuario tendrá la posibilidad de visualizar cuales son las fechas en las que los proveedores suministran a la empresa los pedidos, y por tanto, podrá visualizar que producto llega a la empresa en una fecha y cuál es su proveedor.

Producto	Proveedor	Fecha
AGUJA	Maya Blue	2018-01-03
AGUJA	Navarro Montes	2018-01-03
AGUJA	Hermanos Gil	2018-01-03
AGUJA	Maya Blue	2018-01-10
AGUJA	Navarro Montes	2018-01-10
AGUJA	Hermanos Gil	2018-01-10
AGUJA	Maya Blue	2018-01-17
AGUJA	Navarro Montes	2018-01-17
AGUJA	Hermanos Gil	2018-01-17
AGUJA	Maya Blue	2018-01-24
AGUJA	Navarro Montes	2018-01-24
AGUJA	Hermanos Gil	2018-01-24

Figura 141. Manual de usuario Opción Visualizar relación entre productos y proveedores

Opción Añadir Proveedores

Escogiendo “Añadir Proveedores” del submenú “Gestión de Proveedores”, el usuario tendrá la posibilidad de añadir un nuevo proveedor a la base de datos de la empresa, rellenando todos los campos que se pueden ver en la figura 142.

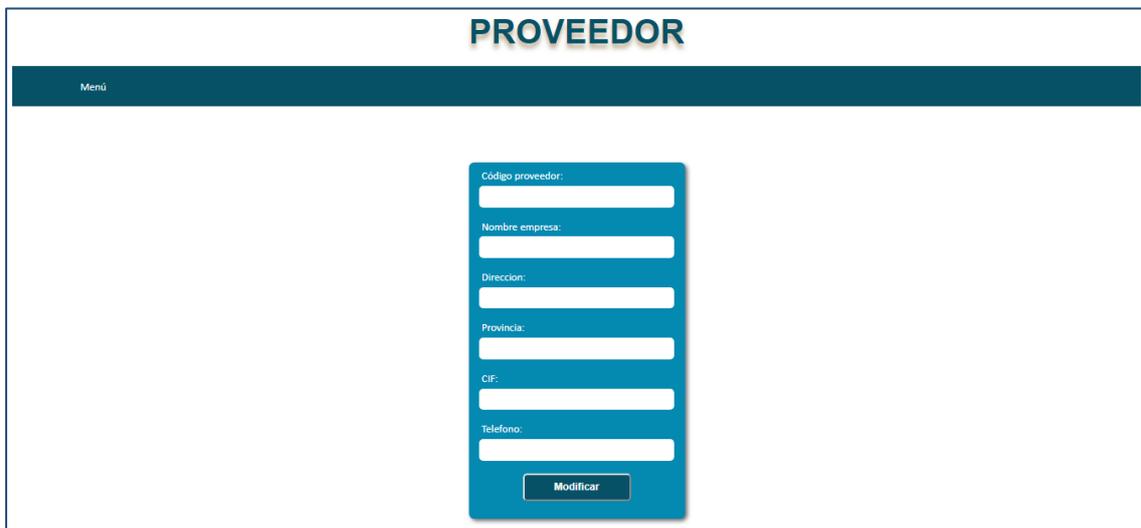


The screenshot shows a web interface titled "PROVEEDOR". At the top, there is a dark blue header with the word "Menú" on the left. Below the header, the main content area is white. In the center, there is a blue-bordered form with the following fields: "Código proveedor:" (with a white input field), "Nombre empresa:" (with a white input field), "Direccion:" (with a white input field), "Provincia:" (with a white input field), "CIF:" (with a white input field), and "Telefono:" (with a white input field). At the bottom of the form is a dark blue button labeled "Insertar".

Figura 142. Manual de usuario Opción Añadir Proveedores

Opción Modificar Proveedores

Escogiendo “Modificar Proveedores” del submenú “Gestión de Proveedores”, el usuario tendrá la posibilidad de modificar un proveedor de la base de datos de la empresa. Aquello que se desee que se mantenga igual dentro de los datos, se dejará vacío de manera que solo se introducirán los cambios nuevos. Lo que es imprescindible de introducir es el código del proveedor.



The screenshot shows a web interface titled "PROVEEDOR". At the top, there is a dark blue header with the word "Menú" on the left. Below the header, the main content area is white. In the center, there is a blue-bordered form with the following fields: "Código proveedor:" (with a white input field), "Nombre empresa:" (with a white input field), "Direccion:" (with a white input field), "Provincia:" (with a white input field), "CIF:" (with a white input field), and "Telefono:" (with a white input field). At the bottom of the form is a dark blue button labeled "Modificar".

Figura 143. Manual de usuario Opción Modificar Proveedores

Opción Eliminar Proveedores

Escogiendo “Eliminar Proveedores” del submenú “Gestión de Proveedores”, el usuario tendrá la posibilidad de eliminar un proveedor de la base de datos de la empresa, simplemente introduciendo el código del proveedor que se desea borrar.



Figura 144. Manual de usuario Opción Eliminar Proveedores

Menú Gráficas

Si el usuario decide seleccionar “Gráficas”, se encontrará un menú desplegable con los diferentes productos a la venta. De forma, que cuando se quiera acceder a la visualización de alguna de las gráficas, con el menú podrá seleccionar el producto deseado y pulsando sobre “Estadísticas” accederá a la gráfica.



Figura 145. Manual de usuario Menú Gráficas

Gráficas

Una vez que el usuario ha seleccionado el producto y pulsado sobre “Estadísticas”, accederá a la evolución de los precios del producto seleccionado a lo largo de las 52 semanas de un año. Asimismo, bajo la gráfica se encontrará una tabla que contiene datos como la semana, el precio antiguo que corresponde al precio que tuvo ese producto la semana anterior, el precio actual y la cantidad en kilogramos que se ha vendido esa semana, facilitando la comprensión del gráfico al usuario.

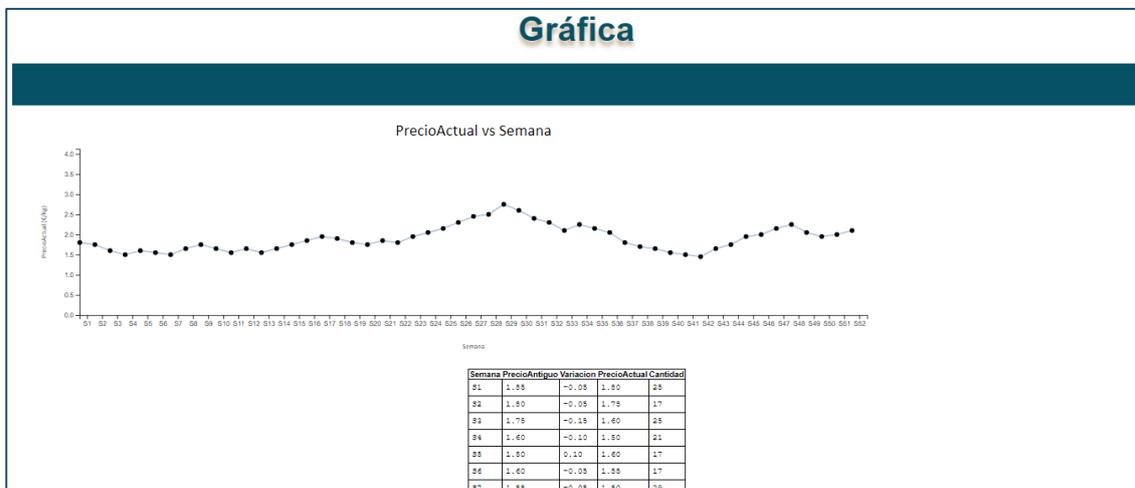


Figura 146. Manual de usuario Gráficas

Nota para el usuario

Para cada una de las pantallas que componen la página web, el usuario tendrá la opción de pulsar el botón “Atrás” que le llevará a la pantalla anterior, o bien la posibilidad de pulsar el botón “Menú” que le trasladará al menú principal de la página web.

Capítulo X

Manual de instalación y planificación

1. Manual de instalación

En la realización del presente trabajo fin de grado, se ha necesitado la instalación de una serie de programas para poder llevar a cabo la creación de la página web, entre ellos se pueden encontrar: Xshell 5, Xftp 5 y Xampp Control Panel.

Xshell 5 – Emulador de Terminal

Es una herramienta diseñada para trabajar en Windows, que ofrece al usuario la posibilidad de establecer conexiones remotas a través de un terminal obteniendo un mayor control sobre los servidores. De forma que a través de diferentes protocolos de conexión el usuario puede conectarse directamente al servidor y tener así acceso directo a todos los archivos y carpetas del sistema del usuario.

Xftp 5

Es una herramienta diseñada para trabajar en Windows, del mismo creador que Xshell 5, que permite interoperar con las herramientas del emulador de terminal, Xshell 5, transfiriendo archivos en la red.

Xampp Control Panel

Es un software libre que consiste en el sistema de gestión de bases de datos MariaDB, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. Esta herramienta de desarrollo permite probar el trabajo desarrollado, en este caso la página web. Además, Xampp incluye PhpMyAdmin para administrar las bases de datos de MariaDB.

2. Planificación del proyecto

La planificación inicial del proyecto se puede ver en la siguiente imagen.

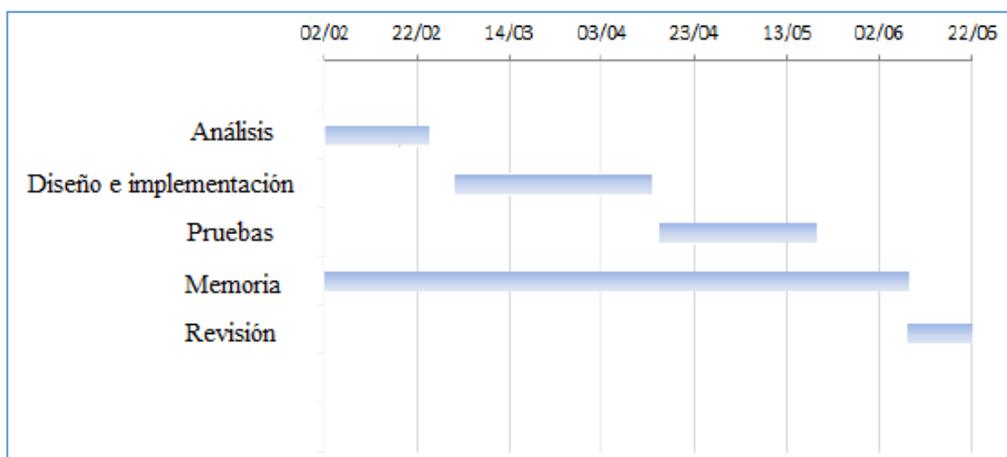


Figura 147. Planificación inicial del proyecto

Como se ha visto en la figura 147, las etapas se encuentran claramente definidas y no existe solapamiento entre ninguna de ellas, de forma que no se empieza una actividad hasta que no ha finalizado la anterior, excepto para la actividad “Memoria” que su realización ha sido a lo largo de la ejecución del resto de tareas excepto para la actividad “Revisión”.

Una vez finalizado el trabajo fin de grado, se puede evaluar la verdadera inversión del tiempo para cada una de las diferentes tareas que ha sido necesario realizar para su consecución.

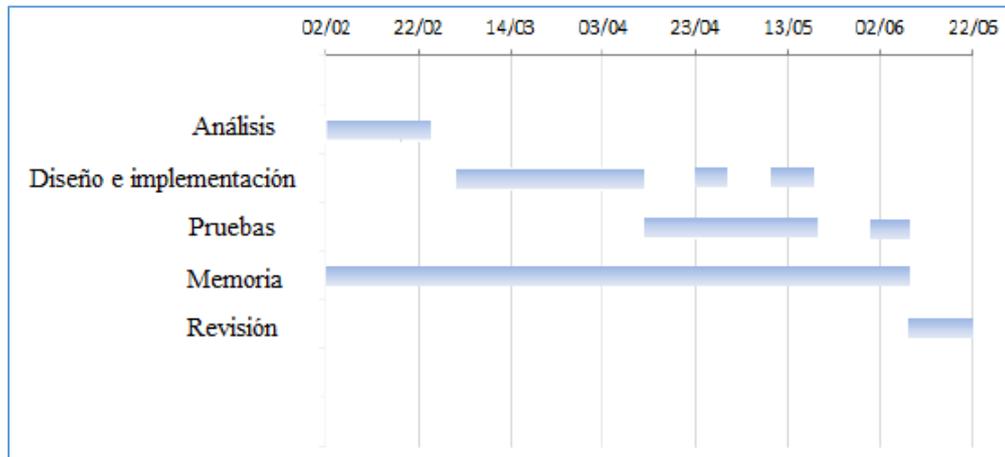


Figura 148. Planificación final del proyecto

Se puede observar que algunas de las tareas que componen el presente trabajo se han alargado más de lo esperado siendo su duración algo intermitente en vez de continua. Esto se debe, fundamentalmente, a la aparición de contratiempos en su realización, como los errores surgidos tanto en los códigos HTML como PHP, los cuales han necesitado más tiempo del esperado, por la dificultad de localizarlos y eliminarlos, así como el tiempo empleado en aprender el lenguaje de JavaScript, pues ha sido algo nuevo que no se ha visto durante la carrera.

Sin embargo, pese a la aparición de contratiempos, por lo general, se ha conseguido el cumplimiento de los objetivos establecidos en la planificación inicial. La idea inicial consistía en no pasar a la siguiente tarea, hasta que no estuviese finalizada la tarea anterior, sin embargo, no siempre ha sido así habiendo ocasiones en las que ha sido necesario trabajar en varias tareas simultáneamente.

Capítulo XI

Conclusiones y líneas futuras de trabajo

En este trabajo se ha realizado una implementación para la Visualización de Datos con D3.js para una determinada empresa. Para ello se ha llevado a cabo el desarrollo de una página web que cumpliera con los requisitos exigidos por la empresa colaboradora del Trabajo Fin de Grado, empleando una máquina virtual cedida por la Escuela de Ingeniería Informática de la Universidad de Valladolid. Se ha validado la página web creada, comprobando que no existía error alguno y que los resultados obtenidos eran los esperados.

Existe evidencia de la gran aportación a la sociedad de la Visualización de Datos, un gráfico correctamente diseñado permite extraer información de un modo intuitivo, sin embargo, si únicamente se cuenta con un conjunto de datos sin representar, la tarea de organizar dichos datos es mucho más costosa. La principal ventaja que ofrece la Visualización de Datos es la capacidad extraer información.

Sin embargo, la Visualización de Datos interactiva da un paso más allá y permite al usuario la creación de gráficos logrando experiencias cada vez más personalizadas. Atrás quedan los tiempos en los que la Visualización de Datos era un simple intercambio de información entre el creador del gráfico y el usuario, en la actualidad va más allá permitiendo al usuario la definición de aquellos datos que desea visualizar y la manipulación convenientemente de los mismos. No obstante, es un área que se encuentra en continua expansión y mejora.

Durante el desarrollo del proyecto, se ha podido dejar evidencia de la gran librería que es D3.js. Su mayor virtud se encuentra en la simplicidad para poder dibujar elementos sencillos como círculos y rectángulos. Además, ofrece un conjunto de funciones predefinidas que permiten al usuario simplificar su tarea en la realización de los gráficos. Otra de sus virtudes es la gran popularidad de esta herramienta, ya que cuenta con una amplia aportación de la comunidad de usuarios, brindando la posibilidad de encontrar una gran multitud de gráficos. En conclusión, D3.js es una librería con una gran utilidad para el usuario y en continua mejora y evolución.

Para concluir con el proyecto fin de grado, se va a plantear algunas líneas futuras que podrían complementar el trabajo desarrollado:

- Aprovechar los datos ya estructurados en las gráficas creadas por esta librería de JavaScript, para la elaboración de diferentes cálculos que permitan la realización de predicciones.
- Implementar todo el entorno web con el que cuenta D3.js en lenguaje de HTML5.

- Mejorar el código empleado en el desarrollo de la página web, incorporando mejoras y avances tecnológicos que facilite en mayor medida la tarea a aquellos usuarios que vayan a hacer uso de la página web.
- Mejoras en el diseño de la página web creada para la empresa colaboradora del Trabajo Fin de Grado.

El ámbito de la Visualización de Datos está en continua investigación y progreso, es por ello que no cesan los avances en esta materia, puesto que es el futuro.

Bibliografía

- [1] A. Sevilla Arias, "Sector Industrial.". Available: <http://economipedia.com/definiciones/sector-secundario.html>.
- [2] S. Ortega Santamaría and Y. Hassan Montero, "Diseño centrado en el Usuario (DCU)". Available: <http://www.nosolousabilidad.com/manual/3.htm>.
- [3] Instituto de Ingeniería del Conocimiento, "7 V del Big Data.". Available: <http://www.iic.uam.es/innovacion/big-data-caracteristicas-mas-importantes-7-v/>.
- [4] "La Visualización de Datos.". Available: <http://www.ingeniovirtual.com/la-visualizacion-de-datos/>.
- [5] A. C. Telea, *Data Visualization: Principles and Practice*, Second. A K Peters/CRC Press, 2014.
- [6] B. Gonzalez del Fresno, "Herramientas para la Visualización de Datos.". Available: <https://www.bbva.com/es/10-herramientas-visualizacion-datos/>.
- [7] E. Meeks, *D3.js in Action*. Manning Publications Co., 2015.
- [8] M. Bostok, "D3.js." [Online]. Available: <https://d3js.org/>.
- [9] "HTML5.". Available: <https://developer.mozilla.org/es/docs/HTML/HTML5>.
- [10] M. A. Alvarez, "JavaScript.". Available: <https://desarrolloweb.com/javascript/>.
- [11] MVP Campus, "Estructura documento HTML5.". Available: <http://www.aprendehtml5.com/html5.htm>.
- [12] MVP Campus, "Modelos de una Base de Datos.". Available: <https://www.campusmvp.es/recursos/post/Disenando-una-base-de-datos-en-el-modelo-relacional.aspx>.
- [13] S. Alvarez, "Paso del modelo E/R al modelo relacional.". Available: <https://desarrolloweb.com/articulos/paso-tablas-entidad-relacion.html>.
- [14] M. Gonzalo Tasis, "Algoritmo de mapeado.". Available: http://cursoanterior2.campusvirtual.uva.es/2016_2017/pluginfile.php/1566286/mod_page/content/10/Tema8_modelologico_2017.pdf.
- [15] DECSAI Universidad de Granada, "Diseño Lógico de Bases de Datos Relacionales." [Online]. Available: <http://elvex.ugr.es/idbis/db/docs/design/5-logical.pdf>.