

Herramientas de Visualización

Tema 1. Introducción a las herramientas de visualización

Índice

Esquema

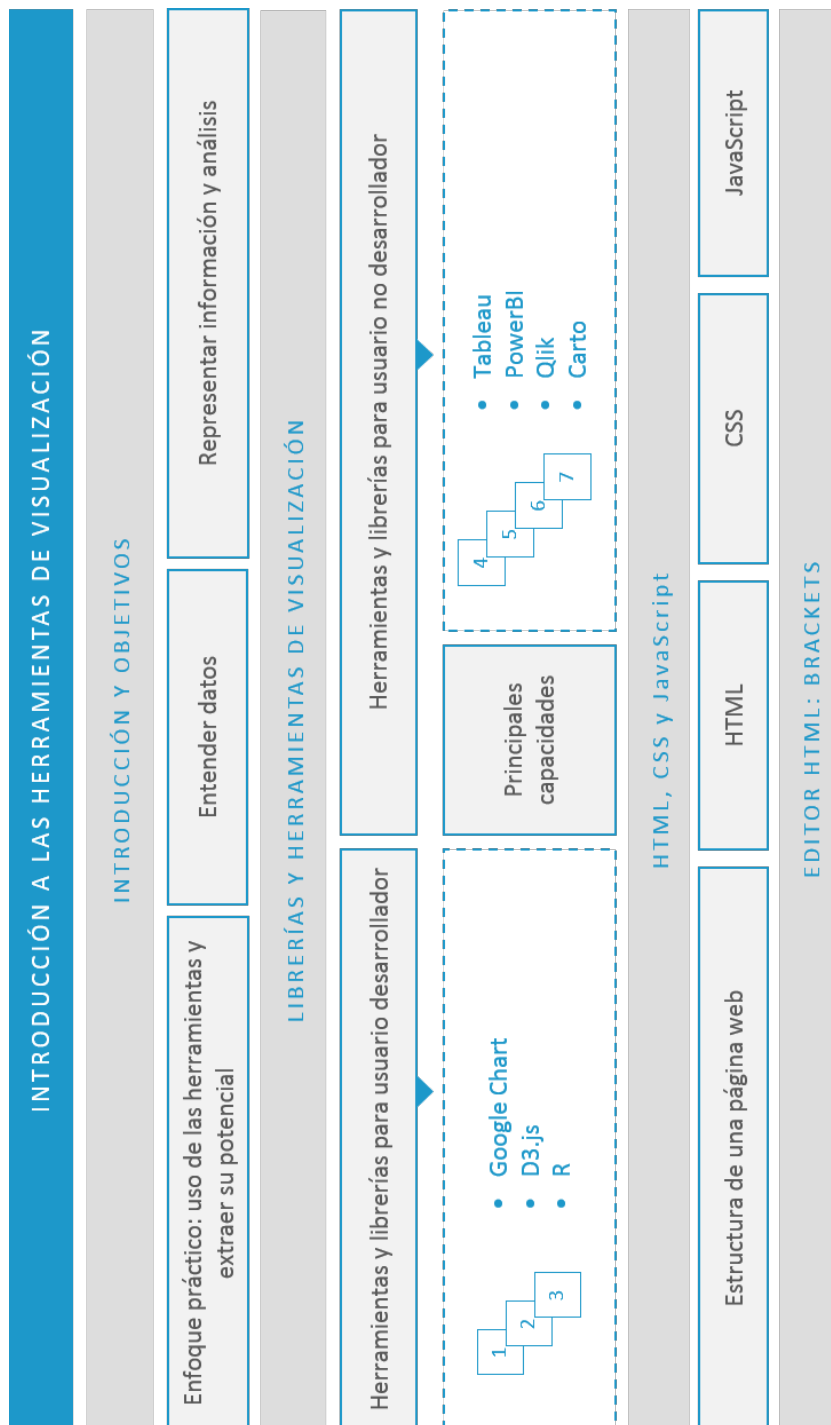
Ideas clave

- 1.1. Introducción y objetivos
- 1.2. Librerías y herramientas de visualización
- 1.3. HTML, CSS y JavaScript
- 1.4. Editor código: Brackets
- 1.5. Referencias bibliográficas

Test

A fondo

- The 27 Worst Charts of All Time
- Graphing guidelines
- Pie charts: the bad, the worst and the ugly



1.1. Introducción y objetivos

El objetivo principal de este tema es que tengas claro cómo has de abordar esta asignatura de Herramientas de Visualización. Esta asignatura tiene un enfoque totalmente práctico en el que abordaremos diferentes librerías y herramientas de visualización y nos adentraremos en su uso y cómo extraer de ellas su mejor potencial.

Adicionalmente, para poder utilizar algunas de dichas herramientas es necesario conocer lenguajes de programación como HTML, CSS o JavaScript. Desarrollaremos los elementos necesarios para poder utilizarlos con solvencia.

En anteriores contenidos ya has tenido la oportunidad de conocer el lado teórico de las visualizaciones y, por lo tanto, aunque podemos discutirlo abiertamente, no se repetirá.

Queremos visualizar datos, explicar una historia con los datos, atraer la atención del lector con un mensaje, pero primero tienes que:

- ▶ Entender por ti mismo qué datos tienes.
- ▶ Ser capaz de representarlos gráficamente con las herramientas adecuadas.
- ▶ Comprobar que los destinatarios entiendan el mismo mensaje que pretendes explicar.

El principal consejo, respecto a la asignatura, es que tú mismo repases todo lo que se va explicando. Lleva su tiempo, pero si intentas de repente entenderlo todo, puedes tener problemas y tendrás que volver al principio. No por correr mucho llegarás antes al objetivo final.

Resumiendo, los **objetivos** de este tema son los siguientes:

- ▶ Familiarizarnos con la asignatura y su metodología.
- ▶ Identificar las herramientas que vamos a utilizar.
- ▶ Aproximarnos a HTML, CSS y JavaScript como lenguajes básicos que utilizaremos en algunas herramientas de visualización.

1.2. Librerías y herramientas de visualización

Los datos y sus relaciones no son siempre algo fácil de entender. Mientras que algunas personas pueden mirar una hoja de cálculo y encontrar intuitivamente la información que necesitan y sus relaciones dentro de una maraña de cifras, otras necesitamos un poco de ayuda, y ahí es donde la visualización de datos puede ser la solución.

La visualización puede ayudarnos a tratar con información más compleja.

No se trata de representar un conjunto de datos de forma amigable y con un diseño elaborado, sino también de hacerlo simplificando la complejidad y destacando aquella información que realmente es relevante, distinguiéndola del resto de los datos.

Una de las mejores formas de transmitir un mensaje es utilizar una visualización para llamar la atención rápidamente sobre los mensajes clave. Y al presentar los datos visualmente, también es posible descubrir patrones y elementos sorprendentes que no serían aparentes con la sola observación de los datos.

Hoy en día, existe una gran variedad de herramientas y recursos que nos permiten hacerlo. Así que para empezar a ayudarte hemos reunido algunas de las herramientas de visualización de datos más destacadas disponibles en la web.

Herramientas y librerías para usuario desarrollador

Para utilizar estas herramientas se requieren **conocimientos de programación**. Esta característica puede suponer una barrera de entrada, pero precisamente este inconveniente es también su mayor ventaja, ya que también les otorga enormes capacidades para personalizar gráficos y prácticamente infinitas posibilidades de

desarrollo.

Google Chart

Es la librería de visualizaciones utilizada por Google Spreadsheets, siendo la API de gráficos de Google. Se trata de una potente y sencilla librería, con una gran variedad de gráficos disponibles.

Accede a la página oficial de Google Chart:

<https://developers.google.com/chart>

D3.js

Es una librería de JavaScript que utiliza HTML, CSS y JavaScript para mostrar algunos diagramas y gráficos sorprendentes de una variedad de fuentes de datos. Esta librería, más que la mayoría, es capaz de algunas visualizaciones muy avanzadas con complejos conjuntos de datos. D3 tiene una poderosa capacidad para gestionar gráficos vectoriales en formato SVG (*Scalable Vector Graphics*).

Accede a la página oficial de D3.js:

<https://d3js.org/>



Figura 1. D3. Fuente: <https://d3js.org/>

R

¿Cuántas otras herramientas tienen un motor de búsqueda completo (<https://rseek.org/>) dedicado a ellas? R es un paquete estadístico enormemente potente, utilizado para analizar grandes conjuntos de datos, aunque con una curva de aprendizaje proporcional a sus elevadas capacidades. Esta librería requiere la utilización de Python como lenguaje de programación.

Accede a la página oficial de R:

<https://www.r-project.org/>

Herramientas y librerías para usuario no desarrollador

La **facilidad de uso** es la característica principal de este grupo de herramientas. Si algo las caracteriza es que es el propio usuario final quien tiene la capacidad para extraer y manejar los datos, crear informes o compartir los datos sin necesidad de pasar por un equipo de desarrollo, reduciendo de esta forma el tiempo que transcurre entre la necesidad y la respuesta. Evidentemente, la autonomía de los usuarios para la visualización de los datos con estas herramientas es máxima.

Según la consultora de mercado Gartner, en la actualidad, las principales plataformas modernas de analítica e inteligencia de negocio (*Analytics and business intelligence*, ABI) ya **no se diferencian por sus capacidades en la visualización de datos**. En su lugar, su diferenciación reside, por un lado, en cómo se integran con el resto de las aplicaciones corporativas de analítica e informes y, por otro, en sus capacidades aumentadas que les permiten acelerar las fases de preparación de los datos y generación de conocimiento para el negocio.

De forma resumida, Gartner evalúa todas las plataformas ABI con base en las siguientes 15 capacidades:

- ▶ **Seguridad:** capacidades que permiten la seguridad de la plataforma, la administración de los usuarios, la auditoría del acceso a la plataforma y la autenticación.
- ▶ **Capacidad de gestión:** capacidades para rastrear el uso, gestionar cómo se comparte la información y por quién, realizar análisis de impacto y trabajar con aplicaciones de terceros.
- ▶ **Cloud:** la capacidad de apoyar la construcción, despliegue y gestión de análisis y aplicaciones analíticas en la nube.
- ▶ **Conectividad de la fuente de datos:** capacidades que permiten a los usuarios conectarse e ingerir datos, estructurados y no estructurados, contenidos en varios tipos de plataformas de almacenamiento.
- ▶ **Preparación de datos:** soporte para la combinación de datos de diferentes fuentes, por medio de arrastrar y soltar, y realizable por los usuarios de negocio, así como la propia creación de modelos analíticos.
- ▶ **Complejidad del modelo:** soporte para modelos de datos complejos, incluyendo la capacidad de manejar múltiples tablas e interoperar con otras plataformas analíticas.
- ▶ **Catálogo:** la capacidad de generar y agregar automáticamente un catálogo que permita la búsqueda de los objetos creados y utilizados por la plataforma.
- ▶ **Insights automatizados:** un atributo fundamental de la analítica aumentada es la capacidad de aplicar técnicas de *machine learning* e inteligencia artificial para generar automáticamente conocimientos para los usuarios finales (por ejemplo, identificando los atributos más importantes de un conjunto de datos).

- ▶ **Análisis avanzado:** capacidades analíticas avanzadas a las que los usuarios pueden acceder fácilmente, ya sea porque estén contenidas en la propia plataforma ABI, o bien porque puedan utilizarse mediante la importación e integración de modelos desarrollados externamente.
- ▶ **Visualización de datos:** soporte para cuadros de mando interactivos y la exploración de datos mediante la manipulación de imágenes gráficas. Se incluye una serie de opciones de visualización que van más allá de las de los gráficos circulares, de barras y de líneas, como son los mapas de calor y de árboles, mapas geográficos, gráficos de dispersión y otros visuales de propósito especial.
- ▶ **Consulta de lenguaje natural:** permite a los usuarios consultar los datos utilizando búsquedas basadas en lenguaje natural.
- ▶ **Narración de datos:** capacidad de combinar la visualización interactiva de datos con técnicas narrativas a fin de empaquetar y ofrecer conocimientos de forma convincente y fácil de entender para su presentación a los responsables de negocio.
- ▶ **Análisis integrado:** las capacidades incluyen un SDK con API y soporte para estándares abiertos con el fin de integrar el contenido analítico en un proceso de negocio, una aplicación o un portal.
- ▶ **Generación de lenguaje natural** (*Natural Language Generation*): la creación automática de descripciones lingüísticamente ricas de los conocimientos encontrados en los datos.
- ▶ **Presentación de informes:** la capacidad de crear y distribuir a los usuarios informes de forma programada.

Tomando como referencia estos criterios de evaluación, Gartner elabora anualmente el informe denominado *Magic Quadrant for Analytics and Business Intelligence Platforms*, donde posiciona a las diferentes plataformas y herramientas del mercado.

Figure 1. Magic Quadrant for Analytics and Business Intelligence Platforms



Figura 2. Cuadrante mágico de Gartner sobre plataformas de analítica e inteligencia de negocio. Fuente: <https://www.qlik.com/us/gartner-magic-quadrant-business-intelligence>

En la asignatura abordaremos aquellas herramientas consideradas líderes del mercado según Gartner: Microsoft (PowerBI), Tableau, Qlik... y adicionalmente Carto por su especialización en datos cartográficos.

Tableau

Repleto de gráficos, tablas y mapas, Tableau Public es una popular herramienta de visualización de datos. Los usuarios pueden arrastrar y soltar fácilmente los datos en el sistema y ver cómo se actualizan en tiempo real.

Accede a la página oficial de Tableau:

<https://www.tableau.com/>



Figura 3. Tableau. Fuente: <https://www.tableau.com/es-es>

PowerBI

Es la propuesta de Microsoft que nos permite conectar diferentes orígenes de datos, transformarlos, limpiarlos, relacionarlos y visualizarlos de forma eficiente y sencilla.

Accede a la página oficial de Power BI:

<https://powerbi.microsoft.com/>



Figura 4. Power BI. Fuente: <https://powerbi.microsoft.com/es-es/get-started/>

Qlik

Es una herramienta que se centra en ofrecer al usuario final la posibilidad de extraer y manipular directamente los datos, creando informes sencillos y todo esto sin la necesidad de pasar por un equipo de desarrollo. Es decir, maximiza la facilidad de uso para usuarios no desarrolladores.

Accede a la página oficial de Qlik:

<https://www.qlik.com/>

Carto

Es realmente una herramienta que hay que conocer. La facilidad con la que puedes combinar datos tabulares con mapas es insuperable. Por ejemplo, puedes alimentar un archivo CSV de cadenas de direcciones y lo convertirá en latitudes y longitudes, además, las trazará en un mapa.

Accede a la página oficial de Carto:

<https://carto.com/>

1.3. HTML, CSS y JavaScript

Esta es una asignatura completamente práctica, por lo que a la hora de utilizar las herramientas y librerías que requieren lenguajes de programación, asumiremos que sabes qué es HTML, CSS y JavaScript. En caso de que no tengas conocimiento previo, te recomiendo echar un vistazo a los siguientes tutoriales:

- ▶ HTML: <https://www.w3schools.com/html/>
- ▶ CSS: <https://www.w3schools.com/css/>
- ▶ JavaScript: <https://www.w3schools.com/js/>

No obstante, demos un repaso a la estructura de una página web y a los conceptos más habituales de estos tres lenguajes.

Estructura de una página web

Una página web se construye con base en los siguientes niveles:

- ▶ **Estructura y contenido:** donde se incluyen los elementos de la página que vemos en el navegador y que se componen de textos, imágenes, enlaces, párrafos, tablas, capas, etc.
 - El lenguaje a utilizar es HTML.
- ▶ **Apariencia:** donde se determinan los colores, tipografías, fondos, tamaños, alineación, etc. de todos los objetos presentes en la página web.
 - El lenguaje a utilizar es CSS.

- ▶ **Comportamiento:** aporta el grado de inteligencia necesaria a la página para poder mostrar efectos y animaciones, también realizar validaciones, automatizar tareas, conectar con bases de datos, llamar a APIs (interfaz de programación de aplicaciones), utilizar librerías externas, etc.
 - Existen multitud de lenguajes de programación que nos permiten trabajar tanto en el frontal como en el *backend*, pero a los efectos de su utilización con herramientas de visualización emplearemos JavaScript.

HTML

HTML es un lenguaje de marcas utilizado para el desarrollo de páginas web. Es una forma de codificar con base en **marcas o etiquetas** que contienen información adicional acerca de la estructura del texto o su presentación.

HTML define la estructura y contenido, pero no define su estilo visual.

Para eso se usará CSS.

Un **documento** HTML contiene dos partes principales: la cabecera y el cuerpo del documento.

- ▶ La cabecera (head) suele contener los metadatos del documento.
- ▶ El cuerpo (body) suele contener el contenido que se muestra en la página final. Si tenemos la necesidad de incluir documentos externos como archivos JavaScript y CSS, esto se hará en la cabecera.

Dispones de más información sobre la sección head en:

https://www.w3schools.com/html/html_head.asp

Si quieres profundizar en la sección body utiliza el siguiente enlace:

https://www.w3schools.com/tags/tag_body.asp

Si nunca has intentado trabajar con HTML, entra simplemente en esta dirección:

http://www.w3schools.com/html/tryit.asp?filename=tryhtml_intro

Cambia el contenido de los códigos HTML que aparecen y aprieta el botón de Run.

CSS

CSS nos permite generalizar y definir el estilo de la página HTML que se visualizará en el navegador y de cada elemento HTML de manera exacta. De esta forma, CSS aísla el contenido de la presentación.

De forma similar a JavaScript, el código se ha de introducir entre las *tags*: `<style>` y `</style>`. Seguiremos el ejemplo anterior, ¿qué pasa si añadimos al anterior ejemplo el siguiente CSS?

```
<style>
body{
background-color:#d0e4fe;
}
h2{color:orange;
text-align:center;
}
p{
font-family:"Times New Roman";
font-size:20px;
}
</style>
```

sobrescribirá el estilo del cuerpo de nuestro HTML. En este caso, estamos definiendo un nuevo color para el fondo de la página. Todos los colores tienen una representación hexadecimal. Si quieres saber a qué color corresponde ese código, edita tu HTML o introduce el código en la siguiente web: <http://html-color-codes.info/>

definirá el nuevo estilo de nuestra cabecera con un nuevo color y alineando el texto en el centro. Podemos observar que el color se define con el nombre del color en lugar del valor hexadecimal. Es una posibilidad para ciertos colores.

definirá el estilo de nuestra fecha. Aquí cambiamos el tipo y tamaño de fuente. Hay muchas más posibilidades que puedes explorar tú mismo. Este es un ejemplo muy básico.

Tanto JavaScript como CSS se pueden definir en otros archivos. Si los definimos en archivos externos, los tendremos que conectar con etiquetas dentro de la cabecera en nuestro HTML (lo que hemos marcado en negrita es lo deberíamos modificar):

- ▶ JavaScript:
- ▶ CSS:

En CSS existen 3 tipos de selectores o identificadores para saber sobre qué elemento HTML debemos aplicar el estilo:

- ▶ De **elemento** HTML: ...
- ▶ De **identificador**: todos los elementos HTML cuya propiedad tenga un determinado valor, tendrán ese estilo.
- ▶ De **clase**: todos los elementos HTML cuya propiedad tenga un determinado valor, tendrán ese estilo.

Dispones de multitud de ejemplos que te ayudarán en el uso de CSS en el siguiente enlace:

https://www.w3schools.com/css/css_examples.asp

JavaScript

JavaScript nos permite añadir un poco de interacción más avanzada. Se suele definir el contenido de las funciones dentro de dos etiquetas: `<button>` y `<p>`. Entre ellas introduciremos el contenido que deseamos. Mejor expliquémoslo con un ejemplo tomado de http://www.w3schools.com/js/tryit.asp?filename=tryjs_myfirst:

En este ejemplo hay dos partes importantes:

```
<button type="button"
onclick="document.getElementById('demo').innerHTML
= Date()">
Click me to display Date and Time.</button>
<p id="demo"></p>
```

Estas partes definirán un botón HTML y qué acción ha de invocar cuando se pulsa en dicho botón. Si vemos el contenido `<button>`, vemos que contiene el nombre de la función `document.getElementById`. El `id` contiene el destino donde se insertará el resultado de la acción.

Expliquemos por partes el código:

- ▶ `document`: corresponde al elemento que define la página HTML en JavaScript. Es decir, cuando escribimos `document` es porque pretendemos realizar alguna acción con el contenido HTML.
- ▶ `getElementById`:
 - Como hemos visto antes, `getElementById` define el destino. Si miramos cómo se definía, veremos que contenía un `id`, que es el mismo que encontramos entre paréntesis después de `getElementById`. Lo único que estamos diciendo con esta función es que vamos a trabajar con el contenido de ese `id`.
 - Además, incluyendo `innerHTML` estamos indicando que vamos a trabajar con el contenido HTML.
 - `Date` es una función definida ya en JavaScript que nos da la hora y la fecha en

el momento en que se invoca.

Es decir, lo que se muestra en este ejemplo es un botón que, al pulsarlo, nos introducirá la fecha y hora actual en la página mostrada por el navegador.

A continuación explicaremos Brackets, donde podremos experimentar con JavaScript, crear nuevos archivos y conectarlos.

1.4. Editor código: Brackets

Brackets es un editor de código fuente abierto, gratuito y creado por la empresa Adobe Systems, orientado al desarrollo web. Está enfocado hacia el desarrollo web de *frontend* (HMTL, CSS y JavaScript).

Accede a la página oficial de Brackets:

<http://brackets.io/>

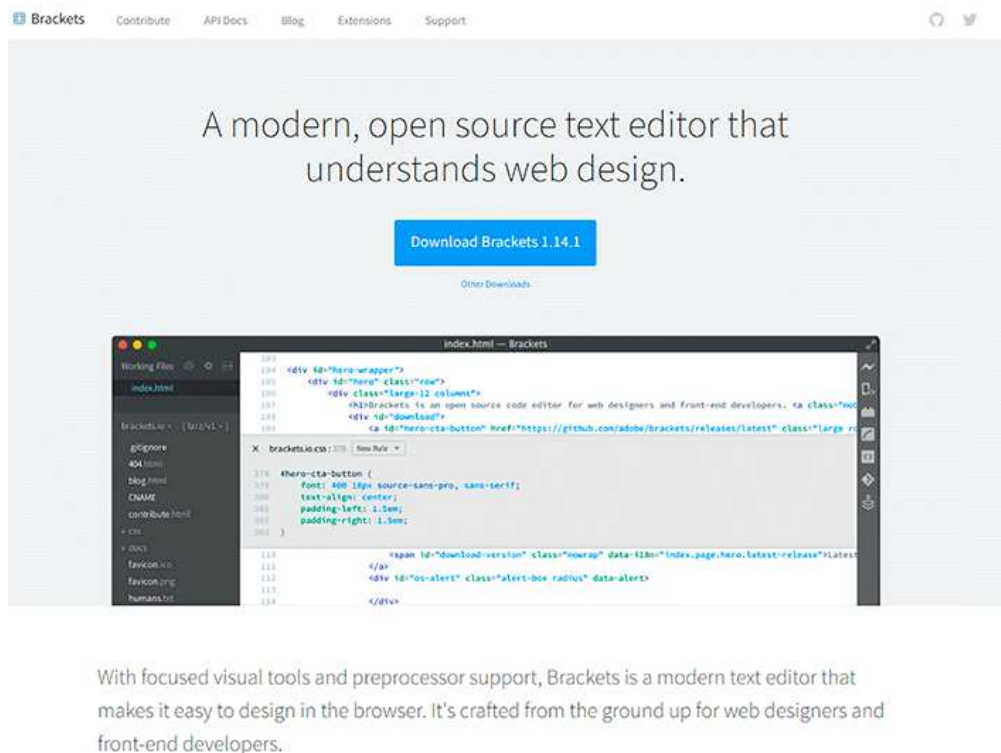


Figura 5. Brackets. Fuente: <http://brackets.io/>

Se trata de una herramienta *open source* con una interfaz sencilla y minimalista, que incorpora ayudas en la escritura del código y sangrado en las sentencias de código, lo cual permite que este sea mucho más legible, todo ello con una organización de todos los elementos del proyecto web con base en proyectos.

Otra facilidad incorporada en Brackets es que utiliza un código de colores bastante visible. Es decir, cuando haces clic en cualquier etiqueta HTML, se marca en color verde tanto el principio como el fin del párrafo, siempre que esté bien cerrado. Si no es así, el resto de etiquetas posteriores las marca en rojo. Esta facilidad ayuda a no cometer errores debido a etiquetas que queden abiertas.

Una de sus características más importantes es el **modo de vista previa en vivo**. En este modo podemos abrir el proyecto web en el navegador y ver en tiempo real las modificaciones que realicemos en el código fuente HTML + CSS + JavaScript. Es una característica enormemente útil porque nos permite modificar cualquier elemento del código y ver reflejados dichos cambios de forma instantánea en la ventana de previsualización, gracias a que incorpora un servidor web local.

Brackets también incorpora la edición rápida de etiquetas. Es decir, si hacemos clic en etiquetas como h1, p, etc., y luego hacemos «Ctrl + E», se despliega el estilo CSS que tenga asociado. Además, nos permite editar dicho CSS desde este punto sin tener que acudir al archivo CSS.

En definitiva, Brackets es una muy buena opción tanto si quieres aprender editando código nuevo como para revisar el código de los ejercicios hechos en clase.

Si necesitas profundizar en este editor, aquí tienes la guía oficial:

<https://github.com/adobe/brackets/wiki/How-to-Use-Brackets>

1.5. Referencias bibliográficas

Brackets: A modern, open source code editor. <http://brackets.io/>

Carto. <https://carto.com/>

D3.js: Data-Driven Documents. <https://d3js.org/>

Gartner. (11 de febrero de 2020). *Gartner Magic Quadrant for Analytics and Business Intelligence Platforms*. <https://www.gartner.com/en/documents/3980852/magic-quadrant-for-analytics-and-business-intelligence-p>

Google Developers. (S. f.). *Google Charts*. <https://developers.google.com/chart>

Microsoft. (S. f.). *Power BI: Microsoft Power Platform*. <https://powerbi.microsoft.com/>

Qlik: Analítica de datos eficaz. <https://www.qlik.com/>

The R Foundation. (S. f.). *The R Project for Statistical Computing*. <https://www.r-project.org/>

Tableau: Software de análisis e inteligencia de datos. <https://www.tableau.com/>

The 27 Worst Charts of All Time

Hickey, W. (26 de junio de 2013). The 27 Worst Charts of All Time. *Business Insider*.
<http://www.businessinsider.com/the-27-worst-charts-of-all-time-2013-6?op=1>

Este artículo nos muestra una serie de gráficos que, aunque innovadores, fueron un fracaso.

Graphing guidelines

Mr. Guch. (18 de marzo de 2012). Graphing guidelines. *Mrphysics*.
<http://mrphysics.org/MrGuch/graph.html>

En este artículo encontrarás algunas directrices para realizar un buen gráfico.

Pie charts: the bad, the worst and the ugly

Frey, L. (5 de febrero de 2013). Pie charts: the bad, the worst and the ugly. *Visuanalyze*. http://visuanalyze.wordpress.com/2013/02/05/bad_piecharts/

En este artículo puedes ver por qué un gráfico circular, uno de los favoritos para la visualización de datos, no siempre es adecuado.

1. Para explicar una historia basada en datos debemos:
 - A. Entender los datos disponibles.
 - B. Ser capaz de representarlos gráficamente con las herramientas adecuadas.
 - C. Comprobar que los destinatarios entiendan el mismo mensaje que pretendes explicar.
 - D. Todas las anteriores.

2. Los datos y sus relaciones no son siempre algo fácil de entender:
 - A. Correcto y por ello las herramientas de visualización nos pueden ayudar a tratar con información compleja.
 - B. Correcto porque los datos siempre requieren simplificar su complejidad.
 - C. Incorrecto si estamos tratando con datos de carácter personal.
 - D. Incorrecto si estamos tratando con datos numéricos.

3. Todas las herramientas de visualización requieren de conocimientos de programación:
 - A. Incorrecto.
 - B. Correcto. Tableau es un ejemplo de ello.
 - C. Correcto. Google Chart es un ejemplo de ello.
 - D. Correcto. D3.js y Carto son un ejemplo de ello.

4. ¿Cuáles de estas herramientas están orientadas a usuarios no desarrolladores?
 - A. Tableau.
 - B. Qlik.
 - C. Carto.
 - D. Todas las anteriores.

5. ¿Cuáles de estas herramientas están orientadas a usuarios desarrolladores?
- A. Google Chart.
 - B. Carto.
 - C. Qlik.
 - D. Todas las anteriores.
6. En la actualidad las principales plataformas modernas de analítica e inteligencia de negocio...
- A. Se diferencian especialmente por sus capacidades en la visualización de datos.
 - B. Ya no se diferencian especialmente por sus capacidades en la visualización de datos.
 - C. Requieren de usuarios avanzados en programación.
 - D. Solo son accesibles por empresas.
7. Enumera, según Gartner, 3 características de las herramientas modernas de visualización de datos la actualidad:
- A. Análisis avanzado, *insights* automatizados y catálogo automático.
 - B. Seguridad, *cloud* y conectividad a fuentes de datos.
 - C. Seguridad, soporte a la preparación de los datos y consulta mediante búsquedas en lenguaje natural.
 - D. Todas son correctas.
8. Una página web se construye con base en los siguientes niveles:
- A. HTML y CSS.
 - B. HTML y editores de código.
 - C. Captura de datos y visualización.
 - D. Estructura, contenido, apariencia y comportamiento.

9. Se dice que HTML es un lenguaje de marcas porque:
- A. Permite las llamadas a APIS o servicios externos de terceros.
 - B. Señaliza todas las líneas de código haciéndolas más legibles.
 - C. Es compatible con otros lenguajes como JavaScript.
 - D. Codifica con base en etiquetas.
10. Brackets es un editor de código:
- A. Compatible con HTML, CSS y JavaScript
 - B. Que incorpora un servidor web local.
 - C. Que permite ver en tiempo real las modificaciones que realicemos en el código fuente.
 - D. Todas son correctas.