ICONOS

Instituto de Investigación en Comunicación y Cultura

DESARROLLO DE UNA GUÍA PARA LA ELECCIÓN DE GRÁFICOS, TIPOGRAFÍA, PALETAS DE COLOR Y LIBRERÍAS JAVASCRIPT PARA LA VISUALIZACIÓN DE DATOS EN LA WEB

TESIS

QUE PARA OBTENER EL GRADO DE

MAESTRÍA EN COMUNICACIÓN CON MEDIOS VIRTUALES

PRESENTA:

JESÚS ENRIQUE DÍAZ AGUILAR

ASESOR:

ELENA VARGAS

Ciudad de México

(Agosto de 2017)

RECONOCIMIENTO DE VALIDEZ OFICIAL DE ESTUDIOS DE LA SECRETARÍA DE EDUCACIÓN PÚBLICA SEGÚN ACUERDO NO. 2005625 DE FECHA 22 DE SEPTIEMBRE DE 2005. Clave 2012

Índice

Introducción

Capítulo 1

Selección de gráficas, tipografías y paletas de colores para la visualización de datos

1.1 Criterios para seleccionar el tipo de gráfica

1.2 Tipografía para la visualización de datos

1.3 Paletas de colores para la visualización de datos

Conclusiones

Capítulo 2

Elección de una librería JavaScript para la visualización de datos

2.1 Librerías JavaScript pensando en un perfil de diseñador Web

2.1.1 Librería High Charts

2.1.2 Librería AM charts

2.1.3 Live Gap Charts

2.2 Librerías JavaScript pensando en un perfil de desarrollador Web

2.2.1 Librería D3.js

2.2.2 Librería AnyChart.js

2.2.3 Librería Charts.js

Conclusiones

Capítulo 3

Desarrollo del sitio Web "Data Picker Guide"

3.1 Mapa de Sitio

3.2 Sección Acerca

3.3 Sección Gráficos

3.4 Sección Colores

3.5 Sección Tipografía

3.6 Sección Librerías

3.7 Sección Recursos

Conclusiones

Conclusiones Finales

Fuentes de Consulta

Introducción

Actualmente, la visualización de datos es una herramienta poderosa que ayuda a diseñadores y desarrolladores Web a transmitir información mediante gráficas, tablas, mapas, etc. Sin embargo, una visualización de datos puede decaer por cosas tan comunes como elegir incorrectamente los gráficos que nos ayuden a transmitir la información, o elegir incorrectamente la paleta de colores o las tipografías que se van a utilizar.

Otro punto a tomar en cuenta a la hora de realizar una visualización de datos es qué herramienta debemos elegir para realizarla. En la actualidad existen una vasta cantidad de tecnologías disponibles que nos facilitan la tarea de realizar las visualizaciones de datos.

Para está investigación se ha optado por elegir solo tecnologías basadas en el lenguaje de programación JavaScript.

JavaScript es uno de los lenguajes más populares para programar en la Web, además, existen una gran cantidad de librerías disponibles basadas en este lenguaje que podemos utilizar para crear visualizaciones de datos. Sin embargo, una de las interrogantes que surgen a la hora de elegir una librería es ¿cuál debo elegir y por qué?, para responder está pregunta es necesario tener una serie de consideraciones, por ejemplo si la curva de aprendizaje para utilizar cierta librería es alta o baja, si tienen algún costo, si ofrecen tutoriales o repositorios de consulta, etc. Por otro lado, nos plantea la necesidad de una guía que nos muestre las características de dichas librerías así como sus diferentes posibilidades.

Este trabajo pretende servir como una guía que oriente a diseñadores y desarrolladores Web, acerca de la selección de los tipos de gráficas, paletas de colores y tipografías para la visualización de datos en la Web. Además, pretende orientarlos y ayudarlos a seleccionar una librería JavaScript para la visualización de datos.

En el primer capítulo se analizan algunas recomendaciones de diversos autores que nos ayudan a definir qué criterios debemos elegir al momento de seleccionar un tipo de gráfico, mapa etc. También se analizan recomendaciones de autores para definir las tipografías y paletas de colores recomendados para la visualización de datos. Esto con el objetivo de identificar los más adecuados para crear nuestras propias visualizaciones.

Para el segundo capítulo se realiza un estudio comparativo de seis librerías JavaScript, con el objetivo de identificar que librería JS puede ser de mayor utilidad para desarrolladores y diseñadores Web, partiendo de la hipótesis de que una librería JS puede ser más adecuada para un diseñador Web si la librería se enfoca más en lo visual y menos en programar y que la librería JS puede ser más adecuada para un desarrollador Web si se enfoca principalmente en el código.

Para llevar a cabo el estudio se abarcan las principales características que son: si tienen algún costo, si la curva de aprendizaje para utilizarlas es alta, si cuentan con algún soporte técnico o en su defecto alguna comunidad a la que podamos acudir en caso de tener dudas o problemas al utilizarlas, si cuentan con tutoriales o repositorios para aprender a utilizarlas, si los gráficos que generan son *responsive*, la cantidad de gráficos que ofrece cada una y si son *crossbrowser*.

En el último capítulo se describe el proceso para la realizar la guía. También se describe las partes que componen la guía, las cuales se dividen en cuatro principales secciones que son: elección de gráficos, elección de tipografía, elección de paleta de colores y el análisis de las librerías JavaScript que fueron seleccionadas para la investigación.

Capítulo 1

Selección de gráficas, tipografía y paletas para la visualización de datos

Algunas de las razones más comunes por las que una visualización de datos se ve opacada son por elegir incorrectamente cosas tan comunes como el tipo de gráfica más adecuada, la tipografía o la paleta de colores.

El primer problema parte de elegir incorrectamente el tipo de gráfica que nos sirve para transmitir de manera más adecuada nuestra información. Para elegir correctamente un tipo de gráfico, debemos partir primeramente de qué acción es lo que queremos dar a entender.

Una vez que tenemos definido el tipo de gráfico, se requiere buscar una tipografía agradable para los usuarios y así como una paleta de colores adecuada que garantice el entendimiento de la visualización.

El objetivo de este capítulo es identificar los tipos de gráficas, tipografías y paletas de colores se recomiendan para la correcta visualización de datos.

Por ello se planteó la pregunta ¿qué criterio se debe tomar en cuenta para la elección de gráficos, fuentes y paleta de colores en la visualización de datos? Y se planteó la siguiente hipótesis, Una vez elegido el tipo de gráfica, partiendo de los principales criterios que son: comparación, composición, distribución y relación, se puede elegir una tipografía que sea de fácil lectura para el usuario y que cargue en todos los navegadores. Posteriormente se podrá elegir una paleta de colores que incluya una degradación de color. Para ello primero se aborda qué tipos de gráficos engloban los criterios antes mencionados, después se exponen el tema de la tipografía para indicar las fuentes que son de fácil lectura para los usuarios y que carguen de manera correcta en casi cualquier navegador.

Por último se plantea qué paleta de colores beneficia la comprensión de los datos.

1.1 Criterios para seleccionar un tipo de gráfica

Los gráficos son herramientas que ayudan a visualizar la información de manera fácil y sencilla, sin embargo uno de sus principales problemas es elegir el adecuado. Existen diferentes tipos de gráficos, dentro de los más utilizados destacan los gráficos de barras, los de pie, los histogramas, los de burbujas, etc.

En el libro, Advanced Presentations by Design (Abela 111-116), del doctor Andrew Abela, nos recomienda categorizar los criterios principalmente en cuatro: relación, distribución, comparación, composición. Una vez seleccionado el criterio, Abela menciona también que se debe tomar en cuenta si buscamos transmitir información estática, o que cambia con el paso del tiempo.

A continuación, se explican los diferentes criterios que plantea Andrew Abela, además se describen los tipos de gráficos que nos recomienda utilizar dependiendo del criterio seleccionado.

Relación

Si se busca transmitir información que relacione dos variables, entonces el gráfico más útil será un gráfico de dispersión, por el contrario, si deseamos transmitir información que contenga tres o más variables, un gráfico de burbuja será más adecuado para nuestro fin.

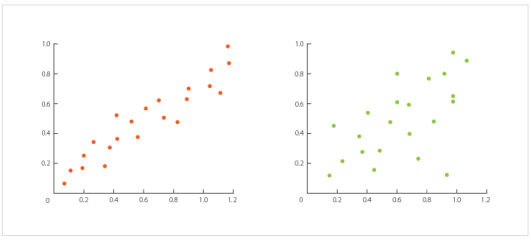


Gráfico de dispersión

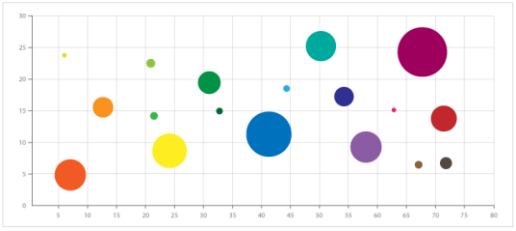
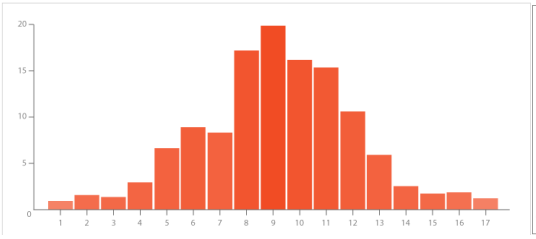


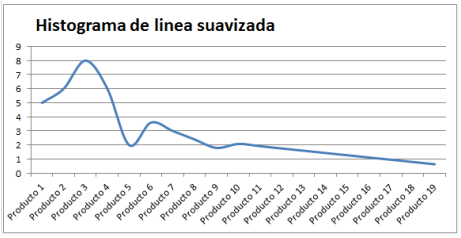
Gráfico de burbujas

Distribución

Si lo que se busca es transmitir la distribución de nuestra información, en caso de que nuestros datos tengan un solo intervalo, se recomienda utilizar un histograma de columna, en caso de que queramos transmitir más de un intervalo se recomienda un histograma de línea. Si queremos mostrar una distribución de dos variables, el gráfico recomendado es un gráfico de dispersión. En caso de que nuestra información tenga tres variables o más el gráfico más adecuado será un gráfico de área.



Histograma de columna



Histograma de línea

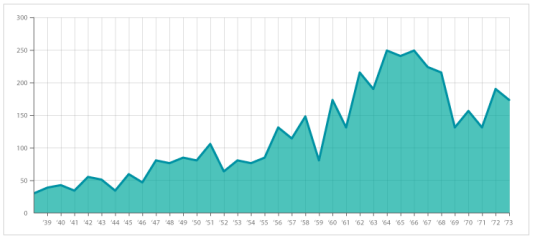
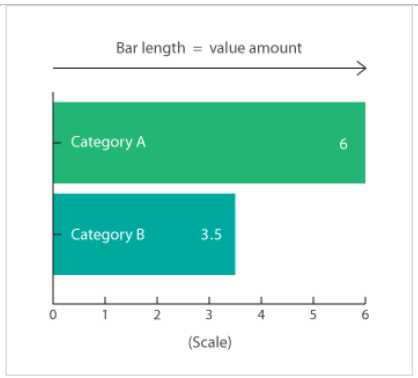
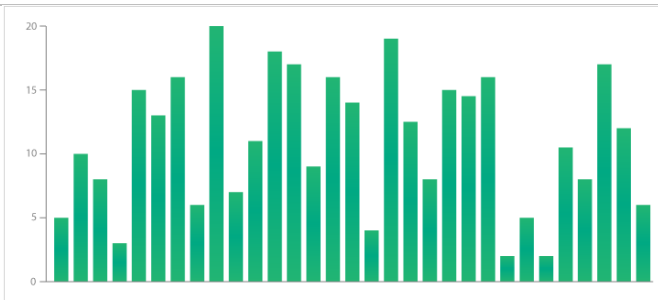


Gráfico de área

Comparación

En caso de que busquemos comparar información, Abela nos recomienda subdividir nuestros criterios para poder elegir mejor el gráfico adecuado. La recomendación nos dice que dividamos en dos: datos estáticos y datos que cambian con el tiempo.

Comparación - Datos Estáticos

Para información que contenga pocos datos es recomendable utilizar un gráfico de barras, en caso que nuestra información posea muchos datos es recomendable usar un gráfico de columnas.	
Para información estática que contenga dos variables, se recomienda utilizar un gráfico de columnas de ancho, donde cada columna representará una variable.	

Comparación - Datos que cambian con el tiempo

Para el caso de comparar datos que cambian con el tiempo existen dos posibilidades que se deben tomar en cuenta, que es la longitud del periodo de tiempo. En caso de querer comprar pocos periodos de tiempo, es recomendable utilizar un gráfico de columnas, si se tienen pocas categorías que se van a comparar. Si contamos con muchas categorías a comparar, un gráfico de línea es una mejor opción.

En caso de que queramos comparar muchos periodos de tiempo, la gráfica de línea también será de utilidad para datos que no sean cíclicos. En caso que nuestra información contenga periodos de tiempo cíclicos, un gráfico "araña" es la mejor opción.

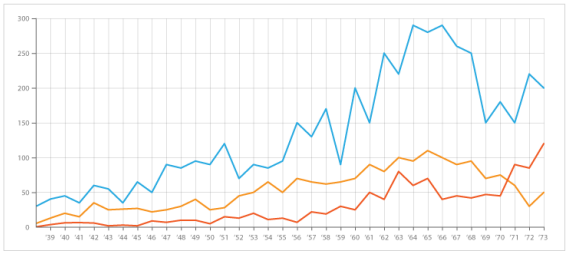


Gráfico de línea

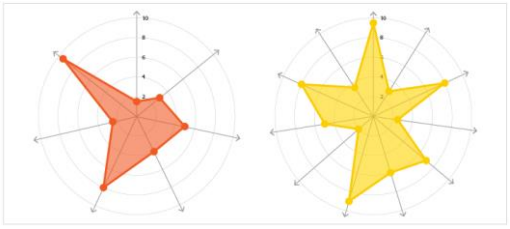
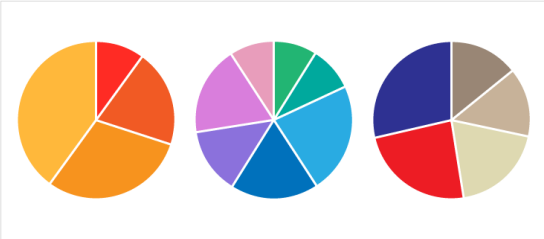
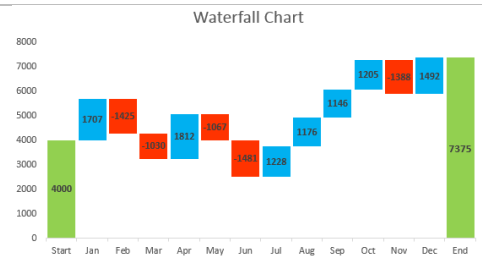
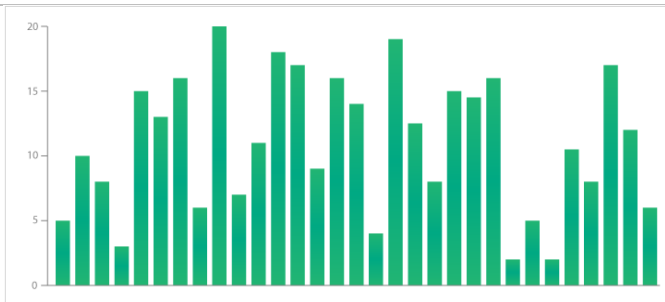


Gráfico de radar o de "araña"

Composición

La composición de los datos se utiliza, cuando se quiere resaltar los componentes de nuestra información. Para este criterio de nuevo debemos elegir si nuestros datos son estáticos o cambian con el tiempo.

Composición - Datos estáticos

Si solo se quiere mostrar porciones de un total, el gráfico de pie es una buena opción, sin embargo, Abela menciona que esta opción puede ser menos efectiva que usar otro gráfico.	
Si se quiere mostrar como los componentes de nuestra información agregan o quitan datos de un total, un gráfico de cascada es el ideal para este objetivo.	
Por último si se requiere mostrar los subcomponentes de un total, un gráfico de columnas apiladas es la mejor opción.	

Composición - Datos que cambian con el tiempo

Si nuestra visualización requiere mostrar de datos que cambian con el tiempo, tomando en cuenta que solo se quieren mostrar pocos periodos de tiempo, un gráfico de columnas apiladas es la mejor opción.

Caso contrario de una visualización que requiere mostrar muchos periodos de tiempo, es preferible utilizar un gráfico de línea en vez de uno de columnas. Para mostrar diferencias relativas y absolutas en el paso del tiempo una gráfica de área apilada es la opción ideal.

A continuación, una imagen que muestra el general de los 4 tipos propuestos:

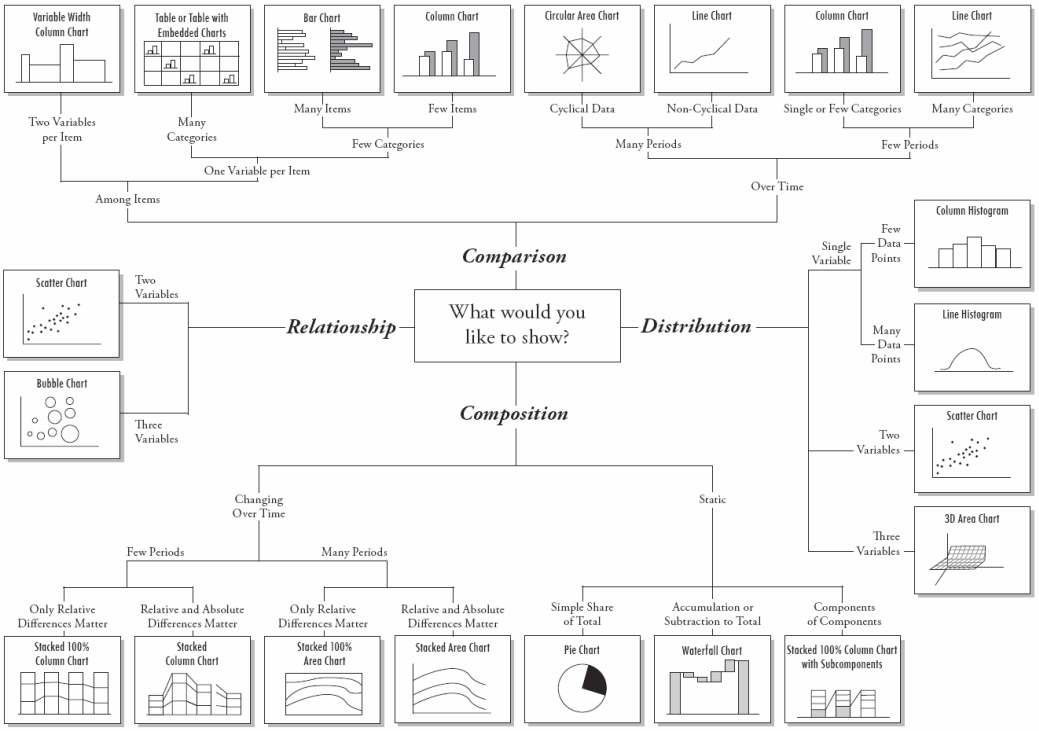


Imagen 1.1 Diagrama realizado por Andrew Abela, tomando como base 4 criterios de selección, con el objetivo de guiar a los usuarios en la elección de una gráfica para la visualización de datos URL.

El diagrama para elegir un gráfico de Andrew Abela es solo una recomendación, el autor nos menciona que lo ideal es probar varios tipos de gráficos antes de seleccionar el gráfico final con el objetivo de visualizar la información de diferentes formas.

Otras dos herramientas disponibles que nos pueden ayudar a dividir por criterios la selección de los gráficos son la guía de Severino Ribecca The Data Visualization Catalogue (Ribecca 2014) y la sección de la página Web de Ann K. Emery Chart Chossing Tool (Emery 2015).

En The Data Visualization Catalogue, Severino Ribecca, nos muestra 16 criterios que podemos tomar en cuenta para elegir el gráfico más adecuado.

Debido a que la investigación se centra en los gráficos que pueden mostrar información cuantitativa se han decidido omitir seis criterios que tienen que ver en mayor medida con información cualitativa. Los criterios que se omitieron son: conceptos, cómo funcionan las cosas, procesos y métodos, análisis de texto, guía de referencias y jerarquía.

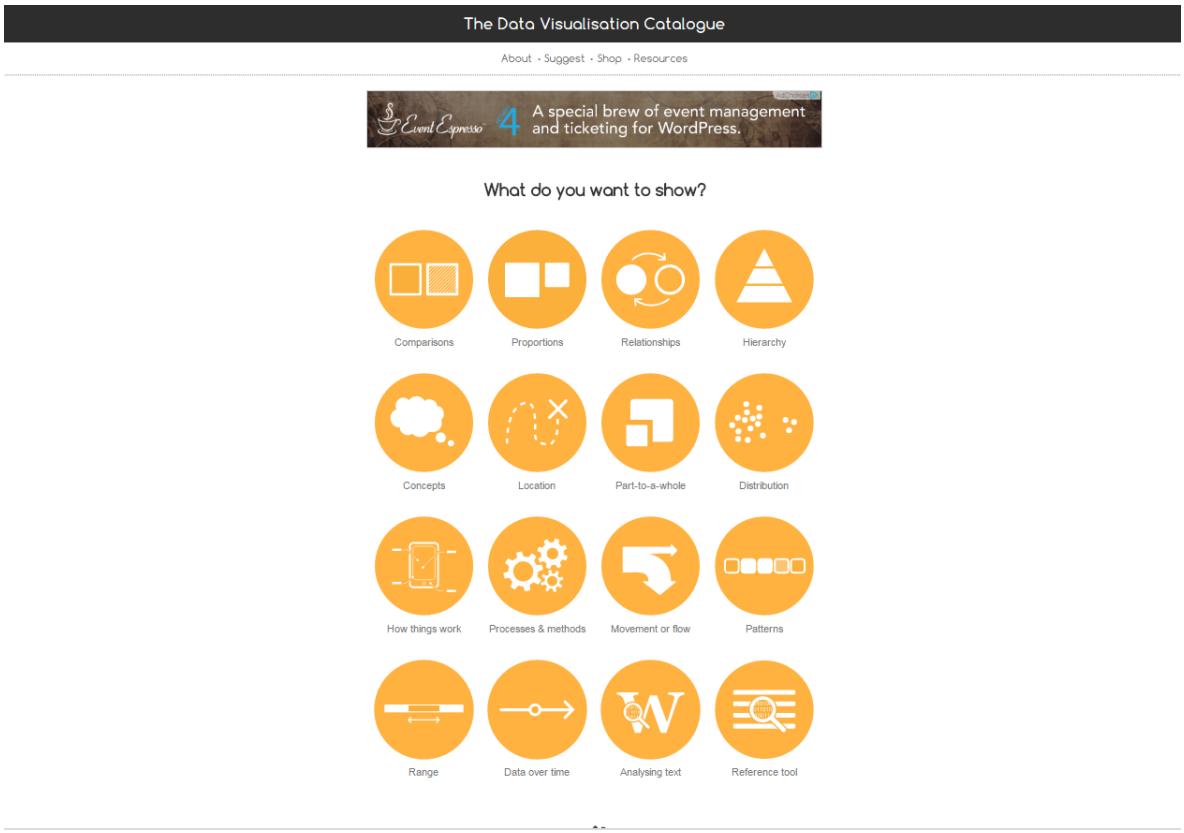


Imagen 1.2 Data Visualization Catalogue, es una guía realizada por Severino Ribecca, para ayudar a seleccionar un gráfico. Consta de 16 tipos de criterios. URL

Comparación

El primer criterio propuesto por Ribecca es el de comparación, y dentro de este criterio nos propone dividirlo en dos categorías: gráficos basados en ejes, y gráficos que no se basan en ejes.

Basados en ejes

Dentro de los gráficos basados en ejes, nos ofrece una opción de 16 gráficos que nos pueden ayudar a la visualización de datos mediante la comparación, en los cuales nos menciona: gráfico de barras, de burbuja, de bala, histogramas, de línea, de columnas apiladas, de rosa, de pirámide, de araña, de radio, de área acumulada, etc.



Imagen 1.3 Gráficos recomendados para el criterio de comparación basado en ejes

URL

No basados en ejes

En los gráficos que nos propone dentro de la categoría de "no basados en ejes" se encuentran gráficos como: gráfico de acorde, mapas coropléticos, gráfico de dona, gráfico de conjuntos paralelos, de pie, de área proporcional, mapa de árbol y diagrama de Venn.

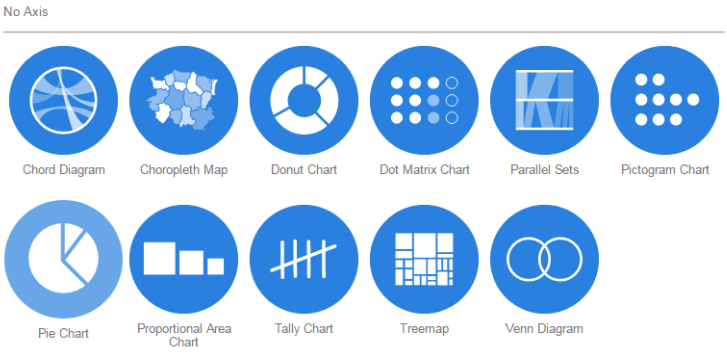


Imagen 1.4 Gráficos recomendados para el criterio de comparación no basado en ejes

URL

Proporción

El segundo criterio que propone es el de proporción dónde se utiliza el tamaño o el área para mostrar las similitudes o diferencias entre los valores o los valores como parte de un todo. Ribecca también propone separar las proporciones en dos categorías: proporción entre valores y proporción como parte de un todo.

Dentro de la proporción entre valores nos propone gráficos como: gráfico de burbujas, mapa de burbujas, gráfico de rosa, de área proporcional, y gráfico de columnas apiladas.

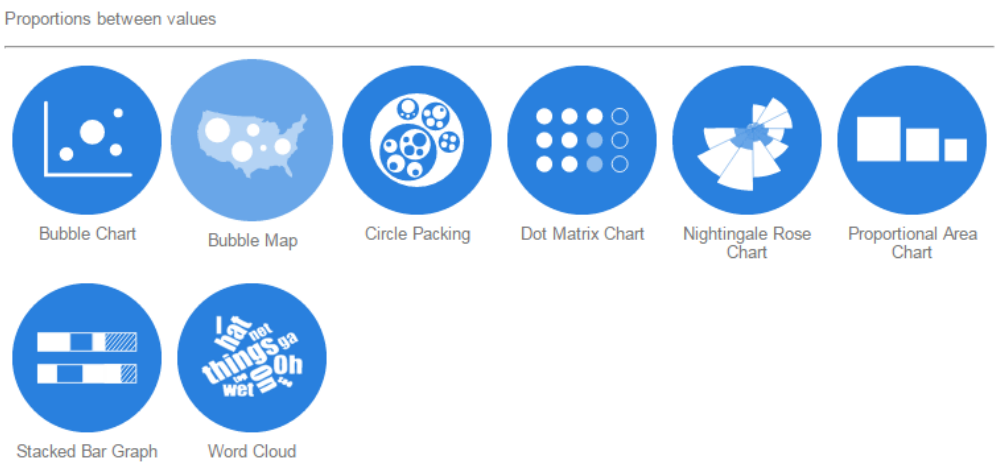


Imagen 1.5 Gráficos recomendados para el criterio de proporción URL

Por su parte en la categoría de proporción como parte de un todo, nos propone gráficos como: gráfico de dona, de columnas apiladas, de pie, diagrama de Sankey, gráfico de columnas apiladas, y mapa de árbol.

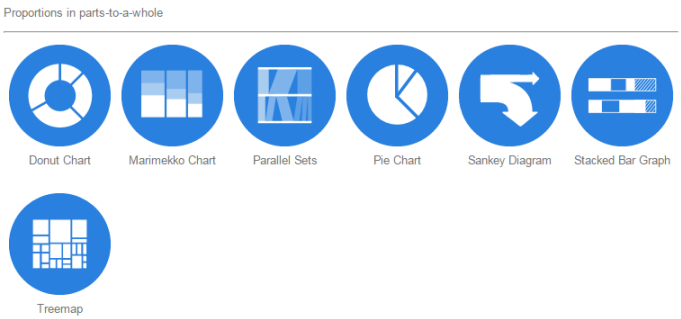


Imagen 1.6 Gráficos recomendados para el criterio de proporción como parte de un todo URL

Parte de un todo

El siguiente criterio que menciona el autor es del de "Parte de un todo". Este criterio es similar al criterio de composición que mencionó el autor Andrew Abela en el primer diagrama que se analizó. En este criterio nos

recomienda utilizar los diferentes gráficos: de dona, de pie, de columnas apiladas, de barras apiladas, mapas de árbol y diagramas solares.

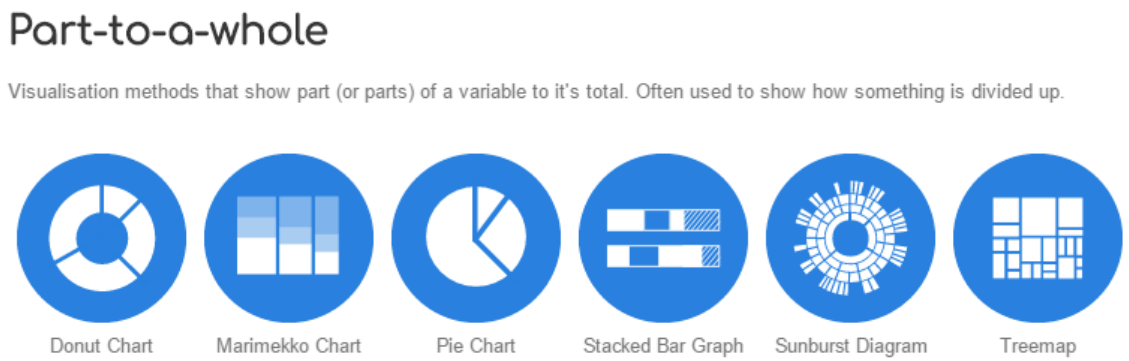


Imagen 1.7 Gráficos recomendados para el criterio de Parte de un todo. URL

Relación

El cuarto criterio que menciona Ribecca es el de relación, y dentro de este criterio nos propone utilizar 16 gráficos, dentro de los que se encuentran: diagrama de arco, gráfico de barras, gráfico de burbujas, gráfico de acorde, mapa de conexiones, gráfico de líneas, de columnas apiladas, diagrama de acordes de cinta, de araña, de radio, de dispersión, de área apilada, de columnas apiladas, diagrama de árbol y diagrama de venn.



Imagen 1.8 Gráficos recomendados para el criterio de relación. URL

Localización

La visualización de datos en Internet nos permite utilizar gráficos más atractivos como son los mapas geográficos. El siguiente criterio se enfoca en la utilización de dichos mapas. El criterio que nos propone Ribecca es el de localización y se nos recomienda utilizar mapas como: mapas de burbujas, coropléticos, de conexiones, de puntos y mapas de flujo.

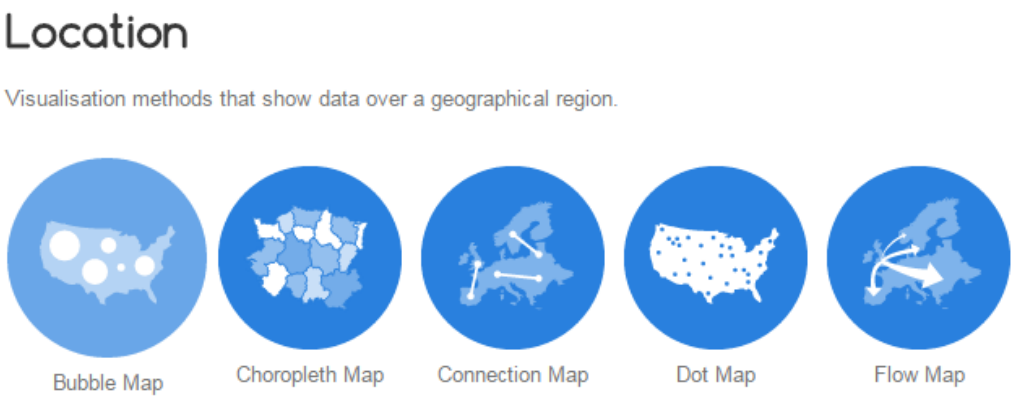


Imagen 1.9 Gráficos recomendados para el criterio de localización. URL

Distribución

Continuando con los criterios de selección, el siguiente es el de distribución. Este criterio pretende mostrar frecuencias, y como la información se extiende dentro de un intervalo o un grupo.

Dentro de los gráficos que nos recomienda utilizar se encuentran: gráfico de cajas y bigote, de burbujas, de densidad, histogramas, de columnas, de conjuntos paralelos, etc.

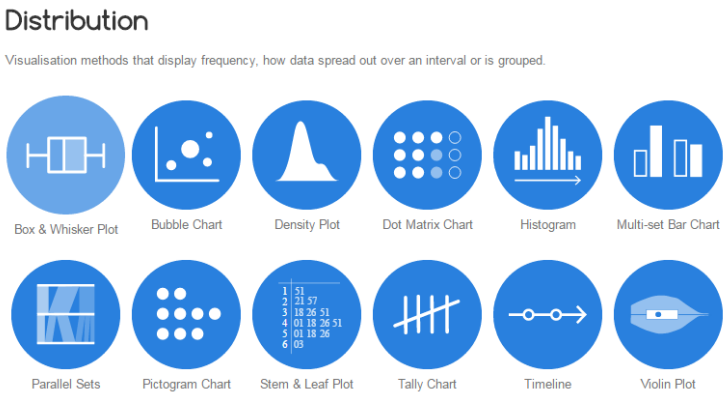


Imagen 1.10 Gráficos recomendados para el criterio de distribución. URL

En el caso que se quiera mostrar distribución basada en un área geográfica nos propone utilizar tres diferentes tipos de mapas: de puntos, de conexión y de flujo.

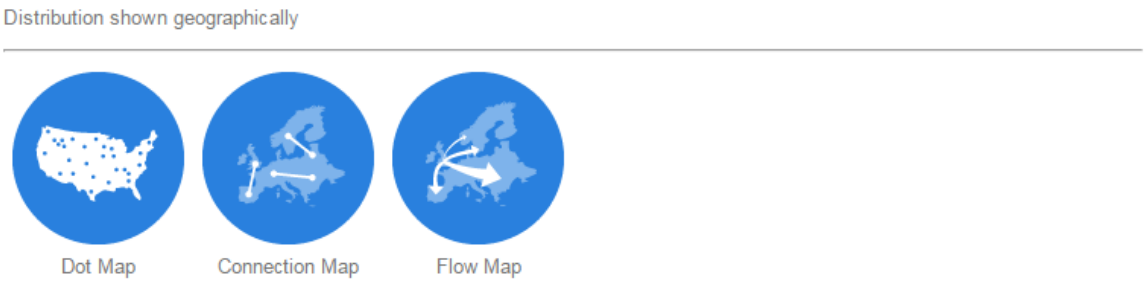


Imagen 1.11 Gráficos recomendados para el criterio de distribución mostrada geográficamente. URL

Movimiento

Este criterio que nos propone pretende mostrar cómo se mueven o fluyen algunos datos mediante mapas geográficos o diagramas.

Dentro de los mapas recomendados a utilizar se encuentran: mapas de conexión y de flujo y los diagramas de Sankey.

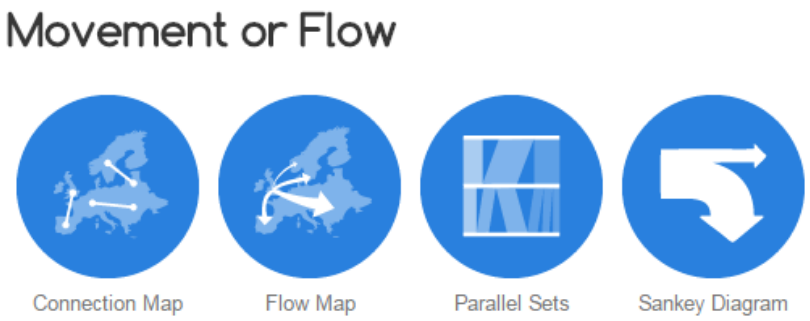


Imagen 1.12 Gráficos recomendados para el criterio de movimiento. URL

Patrones

El criterio de patrones nos propone la utilización de diferentes tipos de gráficos con el objetivo de identificar patrones que nos ayuden a dar significado a los datos. Dentro de este criterio se abarcan muchos de los gráficos descritos anteriormente. Algunos de ellos son: diagramas de arco, de columnas, de cajas y bigotes, de burbujas, mapas corópleticos, de conexiones, de densidad, de puntos, histogramas, de línea, pirámide poblacional, de araña, de dispersión, de espiral, de área acumulada, de columnas apiladas, entre otros.



Imagen 1.13 Gráficos recomendados para el criterio de patrones. URL

Rango

El penúltimo criterio recomendado es el de rango. Dentro de este criterio se busca visualizar las variaciones entre los límites superiores e inferiores dentro de un eje. Dentro de los gráficos recomendados se encuentran: gráfico de cajas y bigotes, de bala, histogramas, etc.

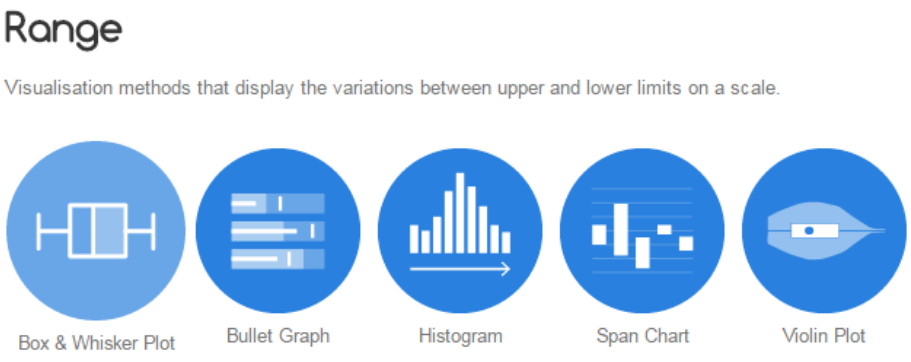


Imagen 1.14 Gráficos recomendados para el criterio de rangos. URL

Información que cambia por periodo de tiempo

El último criterio que nos propone Ribecca en su guía es de la información que cambia por periodo de tiempo. El objetivo de éste, como su nombre lo indica, es mostrar cómo cambian los datos en un periodo particular de tiempo. Dentro de los gráficos recomendados están: los gráficos de área, de burbuja, histogramas, de línea, de rosa, de espiral, de área acumulada, líneas de tiempo, etc.



Imagen 1.15 Gráficos recomendados para el criterio de datos que cambian con el tiempo. URL

The data visualization Catalogue de Severino Ribecca, es una propuesta creada recientemente (2014). Esta propuesta incluye diferentes criterios como opción para ayudar a elegir a los usuarios el tipo de gráficas adecuadas. A diferencia del diagrama de Andrew Abela la guía de Ribecca incluye los mapas geográficos interactivos como método de visualización de datos, además de separar en mayores categorías los criterios de selección que nos pueden ayudar a seleccionar el gráfico más adecuado.

Por último, tenemos la propuesta de Ann k. Emery llamada *Chart Choosing Tool*. En ella Emery nos propone algunos similares para elegir algún tipo de gráfico.

Comparación

El primer criterio que plantea Emery es el de comparación, que ya hemos descrito anteriormente. Dentro de este criterio nos propone algunos gráficos que ya nos han propuesto tanto Abela como Ribecca, como los de barras o de barras acumuladas. También agrega algunos propuestos por ella, las gráficas de lado a lado, o las gráficas de espalda con espalda. Así como la propuesta de gráficas de círculos y de puntos.



Imagen 1.16 Gráficos recomendados para el criterio de comparación. URL

Dispersión

El siguiente criterio que nos menciona Emery en Chart *Choosing Tool,* es otro que ya conocemos anteriormente que es el de dispersión. En éste nos recomienda la utilización de gráficos como histogramas, de pirámide poblacional, de cajas y bigotes y de "*span*".

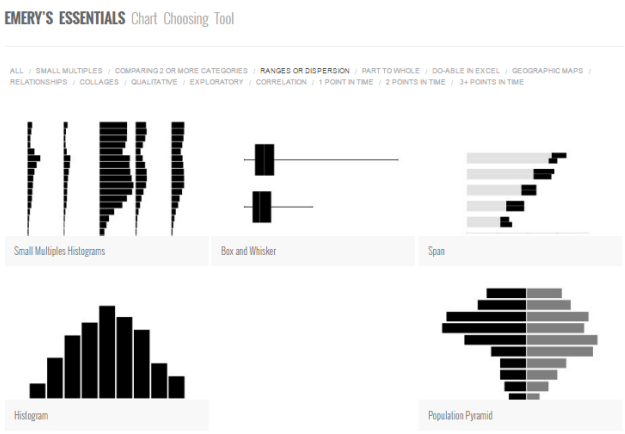


Imagen 1.17. Gráficos recomendados para el criterio de dispersión. URL

Parte de un todo

El tercer criterio propuesto por Emery es el de "parte de un todo". Este criterio podría tomarse en cuenta como el criterio de composición que nos proponía Andrew Abela en su diagrama, ya que su objetivo es similar.

Los gráficos que nos propone utilizar Emery son: de pie, de dona, de columnas acumuladas, y de área acumuladas.

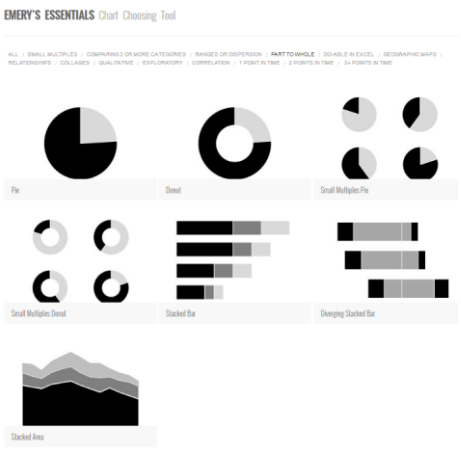


Imagen 1.18 Gráficos recomendados para el criterio de composición (parte de un todo). URL

Mapas Geográficos

En el apartado anterior ya tuvimos un acercamiento con los datos que pueden ser visualizados mediante mapas geográficos. Emery menciona este tipo de gráfico en su herramienta. Sin embargo, solo nos propone los mapas de calor como alternativa.



Imagen 1.19 Gráficos recomendados para el criterio de mapas geográficos. URL

Correlación

El siguiente criterio que nos menciona Emery es el de correlación. En este criterio, propone utilizar los siguientes gráficos: los gráficos de burbujas y de dispersión.

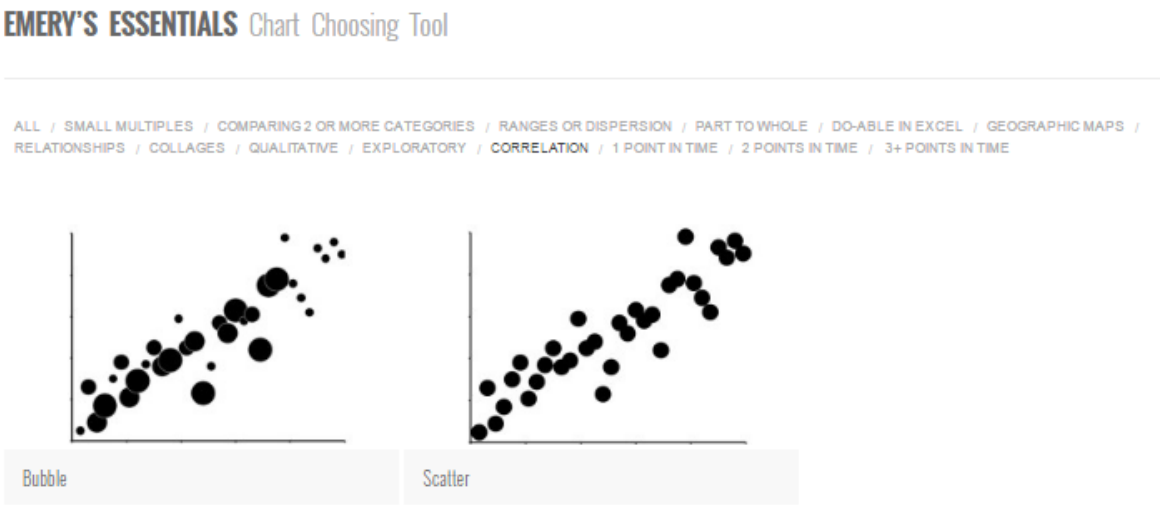


Imagen 1.20 Gráficos recomendados para el criterio de correlación. URL

Datos que cambian en el tiempo

Por último cierra con el criterio de "datos que cambian en el tiempo" y dentro de esta categoría divide en tres categorías los datos: un punto en el tiempo, dos puntos en el tiempo y más de tres puntos en el tiempo. Está categoría es similar a la propuesta por Severino Ribecca en The Data Visualization Catalogue, solo diferenciada por los puntos sobre los periodos de tiempo.

Dentro de la categoría de un punto en el tiempo, Emery nos recomienda una gran cantidad de gráficos como: de pie, de dona, de barras acumuladas, de barras de espalda con espalda, de burbuja, de dispersión, de caja y bigote, mapas de calor, histogramas y puntos entre otros.

En la categoría de dos puntos en el tiempo las recomendaciones de la guía son: gráfico de barras acumuladas, gráficos de barras de espalda con espalda, múltiples mapas de calor, entre otros.

Para la categoría de más de tres puntos en el tiempo las recomendaciones de gráficos son: gráficos de línea, de área, de área acumulada, principalmente.

Una vez concluido el análisis de los tres autores, se decidió crear una tabla que sintetiza los criterios propuestos con el objetivo de identificar los que más se mencionaban. Dicha tabla se muestra a continuación.

	Criterio								
Autor	Comparación	Relación	Composición/Parte de un todo	Distribución	Proporción	Patrones	Rango	Información que cambia con el tiempo	Dispersión
Andrew Abela	x	x	x	X					
Severino Ribecca	x	X	X	X	x	x	x	x	
Ann K. Emery	x	x	x					x	x

Tabla 1. Comparativa de los criterios propuestos por los tres autores

La tabla anterior nos muestra que los criterios de comparación, relación, composición/parte de un todo son mencionados en las tres herramientas. El criterio de composición mencionado por Andrew Abela es mencionado por los otros dos autores, pero bajo otro nombre que es el de parte de un todo. Aunque el nombre cambia la idea muy similar, por lo que se tomó en cuenta para incluirlo en uno solo (composición/parte de un todo). Por

último el criterio de distribución es mencionado por dos de los tres autores analizados.

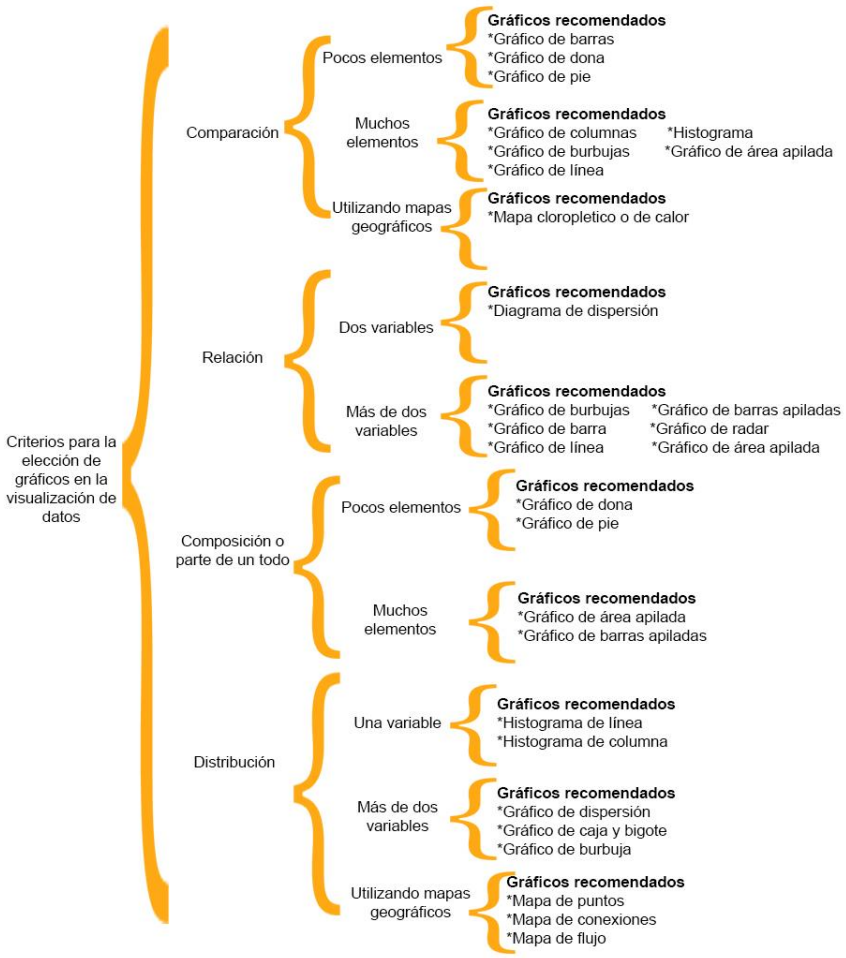
Los tres autores coinciden en tomar en cuenta los periodos de tiempo al realizar algún gráfico. Sin embargo Abela no coincide en tomarlo como un criterio aislado, él menciona a los periodos de tiempo como un subtema dentro de sus cuatro criterios principales.

En las últimas dos guías se puede encontrar un tipo de gráfico que es muy útil que son los mapas geográficos. Estos mapas son usados principalmente en el criterio de comparación, y se utilizan para comparar datos muchos datos que tengan relación con una zona geográfica en específico. Probablemente esto se deba a que el diagrama realizado por Abela fue realizado hace más tiempo y no se acostumbraba utilizarlos.

En las tres herramientas, los autores coinciden en que hay muchos gráficos que pueden servir para diferentes criterios de selección, es cuestión de adecuarlos a nuestras necesidades.

En síntesis, los tres autores nos dan una guía de lo que ellos recomiendan considerar al momento de elegir un gráfico. Los tres coinciden en los criterios de comparación, relación y composición/parte de un todo. Los tres también coinciden que es importante tomar en cuenta si vamos a utilizar datos estáticos o datos que cambian con el tiempo.

Es importante señalar que antes de seleccionar un gráfico lo ideal es realizar pruebas con distintos gráficos, esto con el objetivo de visualizar los datos de diferente manera. De esta forma se tendrá un panorama más amplio para seleccionar el adecuado.



Esquema 1. Criterios y gráficos que se propone utilizar para la visualización de datos

En el esquema anterior se muestran los criterios en los que coinciden los autores revisados. Los autores coinciden en que los criterios de comparación, relación y composición/parte de un todo son los más utilizados al momento de elegir un gráfico. En el criterio de distribución solo coinciden dos autores, pero se decidió incluirlo en el diagrama, porque considero que es un criterio relevante y que puede beneficiar a los

usuarios de la guía conocerlo. Se puede notar que muchos de los tipos de gráficos se repiten dentro de todos los criterios, por lo que se recomienda realizar varias pruebas con diferentes mapas con el objetivo de seleccionar el más adecuado.

1.2 Tipografías para la visualización de datos

Una vez seleccionado el tipo de gráfico mediante algún criterio anteriormente descrito, lo siguiente que debemos hacer es elegir la tipografía que refuerce nuestra visualización de datos.

En el artículo Choosing fonts for Data Visualizations (Yuk-Diamond 2016), de Miko Yuk y Stephanie Diamond, los autores hacen hincapié en tomar en cuenta cinco principales características.

-Poner en un tamaño más grande la información importante

-Ser consistentes (Este punto se refiere a si determinamos un tamaño de letra para los títulos, por ejemplo, 20px, todos los títulos deben ser de 20px).

-Evitar escribir todo en mayúsculas

- Tomar en cuenta que la gente navega en diferentes navegadores

Este punto se refiere a que si hacemos una visualización de datos debemos elegir una fuente que pueda cargar regularmente en todos los navegadores, evitar incluir fuentes que nuestro navegador predeterminado no podrá cargar. Una forma de conocer dichas fuentes es consultar las Safe Web fonts[5], que nos ofrece la página de w3schools.

-Evitar usar tipografías personalizadas

[5] http://www.w3schools.com/cssref/css_Websafe_fonts.asp

Este punto se refiere a que evitemos usar tipografías que de acuerdo a nuestro criterio son legibles. Debemos recordar que cada gente es distinta, y que algunas no están acostumbradas a leer algunas tipografías.

Los autores del artículo recomiendan utilizar Verdana y Garamond como una posible opción de fuentes para nuestras visualizaciones. Eligen estás fuentes porque se encuentran dentro de la lista de Web browser safe fonts, además de poseer un estilo similar a la fuente arial, que es común para la mayoría de las personas.

Por su parte, Howard Coale en el artículo Importance of Color, fonts and icons (Coale 2016), nos menciona que el diseño de la presentación nunca debe demandar nuestra atención. Coale nos dice que en el mundo de la tipografía, debe haber un balance entre la inteligibilidad[8] y la legibilidad[9]. Algunas de las tipografías que logran un balance entre inteligibilidad y legibilidad sin distraer la atención del lector son: Helvetica, Arial y Gill Sans (Coale 25). Aunque el autor menciona que en general las tipografías de la familia sans serif son buenas para utilizarlas en gráficos.

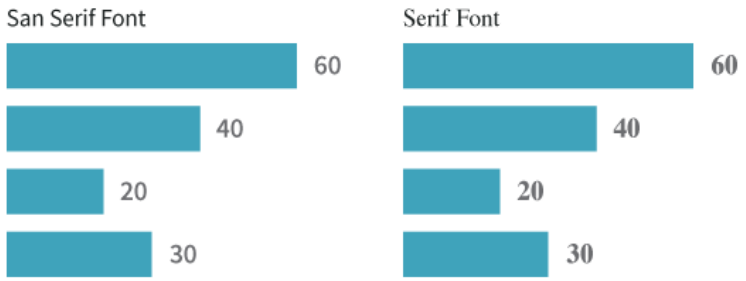


Imagen 1.21 Muestra de familias Sans Serif y Serif

[8] La inteligibilidad es la característica de una tipografía que permite al ojo distinguir un caracter del otro. En algunas fuentes, las formas de algunas letras causan que la tipografía tenga una legibilidad disminuida.
[9] La legibilidad es la relativa facilidad con que una fuente puede ser leída cuando sus caracteres se agrupan en palabras, oraciones y párrafos

Por último, Amy Cesal realizó una guía de estilos llamada Data Visualization Style Guidelines (Cesal 2016), donde nos menciona una serie de recomendaciones enfocadas a la selección de tipografías para la visualización de datos.

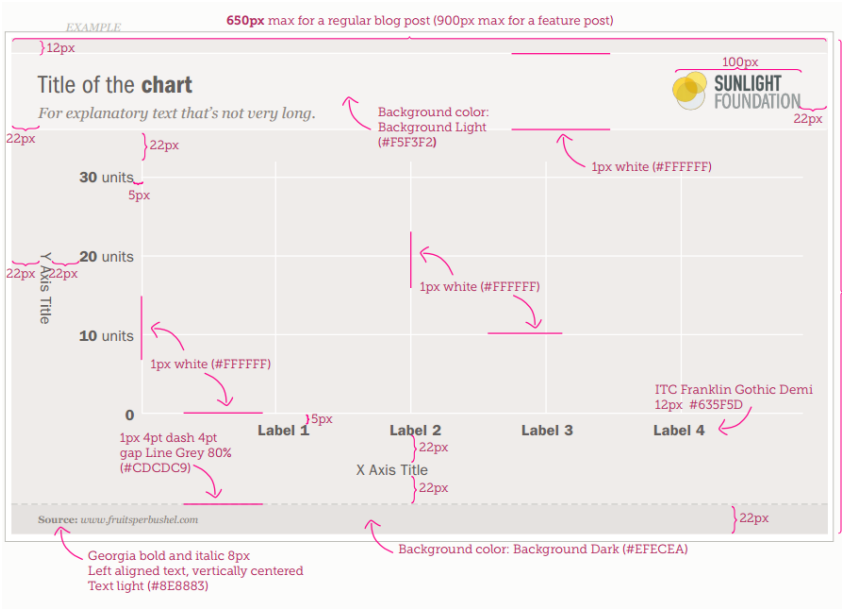


Imagen 1.22 Guía realizada por Amy Cesal dónde ofrece recomendaciones acerca de tamaños de fuentes y medidas de para la visualización de datos. URL

Dentro de las recomendaciones que nos da Amy Cesal, nos recomienda utilizar principalmente dos fuentes que son: ITC Franklin Gothic y Georgia. ITC Franklin Gothic para los títulos, la información en los ejes (en caso de tener) y la información en etiquetas clave.

Por otro lado, nos recomienda utilizar la fuente Georgia en información complementaria y también para citar una fuente de información.

La guía de Cesal va un paso más adelante y nos hace recomendaciones también acerca del tamaño de fuentes que podríamos ocupar para nuestras visualizaciones.

Para el título nos recomienda utilizar un tamaño de 20px y un tamaño de 12px para la información que complementa al título.

En la información de los ejes, nos recomienda utilizar un tamaño de fuente de 12px, pero recomienda usar negritas para la información de las etiquetas principales.

Para información complementaria, Cesal propone utilizar un tamaño de fuente de 10px. Por último, para citar una fuente de información nos recomienda usar un tamaño de fuente de 8px.

Una vez concluido el análisis de los tres autores se decidió realizar una tabla comparativa donde se reunirán las características mencionadas por los autores. Dicha tabla se muestra a continuación.

Características						
Autor	Diferentes tamaños de fuente para la información	Consistencia	No escribir todo en mayúsculas	Safe Web fonts	Evitar las tipografías personalizadas	Fuentes recomendadas
Miko Yuk/Stephanie Diamond	x	x	x	x	x	Verdana, Garamond
Howard Coale	X	X			x	Arial, Helvetica, Gill Sans
Amy Cesal	X	x			X	ITC Franklin Gothic y Verdana

Tabla 2. Tabla comparativa de las recomendaciones que nos proponen los 3 autores

En la tabla anterior, se observa que los tres autores coinciden en utilizar diferentes tamaños de fuente para diferenciar la información que vamos a utilizar. También coinciden en que los tamaños de las fuentes que seleccionemos deben ser consistentes, es decir, debemos utilizar los

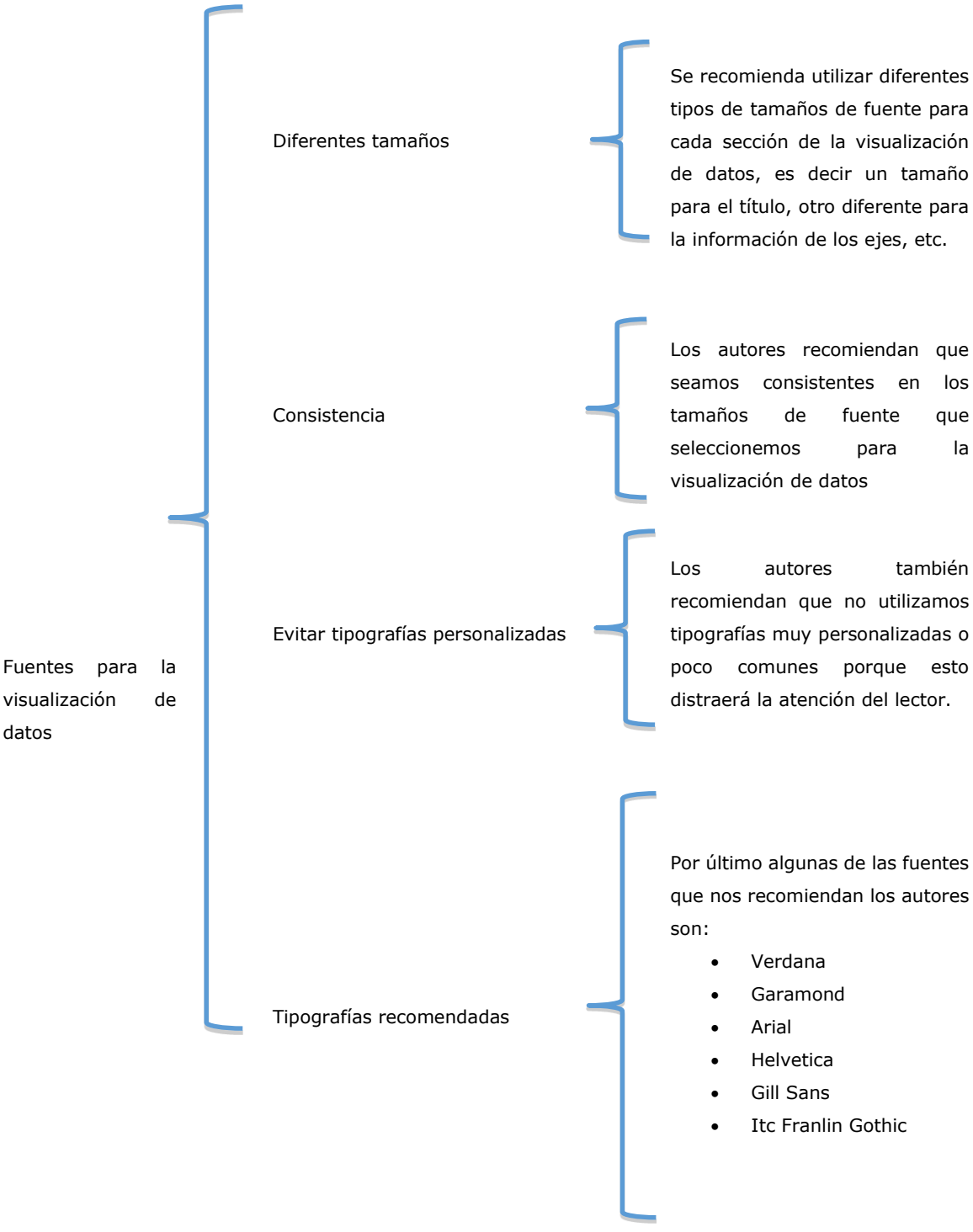
mismos en toda la información que se repita. Amy Cesal es la única autora de los tres que nos recomienda utilizar como medida 20px para los títulos y 12 px para la información complementaria.

Los tres autores coinciden en evitar utilizar tipografías personalizadas que pudieran llegar a confundir al usuario. Las tipografías recomendadas por los autores varían. Por un lado, Miko Yuk y Stephanie Diamond recomiendan solo utilizar tipografías que se consideran Safe Web Fonts, y las dos que recomiendan utilizar son Verdana y Garamond. Howard Coale no menciona que utilicemos solo tipografías Safe Web Fonts, sin embargo, dos de las tipografías que propone están dentro de ellas, que son Arial y Helvetica. Por último, Amy Cesal nos recomienda utilizar ITC Franklin Gothic y Verdana.

Por último, las tipografías propuestas por los tres autores tienen en común que la mayoría son de tipo sans-serif, excepto por la tipografía Garamond que es de tipo Serif.

Como síntesis, se puede decir que los autores recomiendan elegir un tamaño diferente para diferenciar la información principal de la complementaria y ser consistentes con los tamaños seleccionados.

Además de tratar de utilizar fuentes de la familia tipográfica sans-serif, y tratar de evitar las tipografías personalizadas que pudieran llegar a confundir al usuario.



Esquema 2. Esquema que contiene las síntesis de las recomendaciones propuestas por los tres autores

1.3 Paleta de colores para la visualización de datos

> "El color usado correctamente puede mejorar y clarificar una presentación. El color usado pobremente causará confusión"
>
> Maureen Stone.

Ya hemos tocado puntos muy importantes para la realización de una buena visualización de datos. Es el turno de tocar el punto de la elección de paleta de color para la visualización de datos.

Como menciona Maureen Stone, los colores en nuestra visualización pueden ayudarnos a terminar de transmitir correctamente lo que deseamos o terminar por confundir a los espectadores.

En el artículo Finding the right color palettes for data visualization, de Samantha Zhang, la autora nos menciona algunos de los problemas en la elección de la paleta de colores para la visualización de datos. El primero de ellos es que las paletas regularmente propuestas no están diseñadas específicamente para la visualización de datos. Algo que no toman en cuenta estas paletas es la accesibilidad. Especialmente no toman en cuenta los colores para la gente daltónica.

Otro problema recurrente en las paletas de colores recurrentes es que no tienen suficientes colores. Zhang nos dice que una paleta de colores para la visualización de datos debe poseer como mínimo seis colores, para cubrir todos los posibles escenarios.

El último problema que nos dice Samantha Zhang (Zhang 2016), en el artículo es que son difíciles de distinguir. Nos dice que a pesar de que

muchas paletas se basan en gradientes de color, algunas no tienen la suficiente variación de brillo y por ello es difícil para las personas distinguir rápidamente lo que quieren representar.

El artículo finalmente propone tres recomendaciones a seguir al momento de elegir una paleta de color.

La primera recomendación es tener un rango amplio entre el matiz y el brillo de los colores que elijamos. Esto con el objetivo de hacer las visualizaciones accesibles a las personas daltónicas.

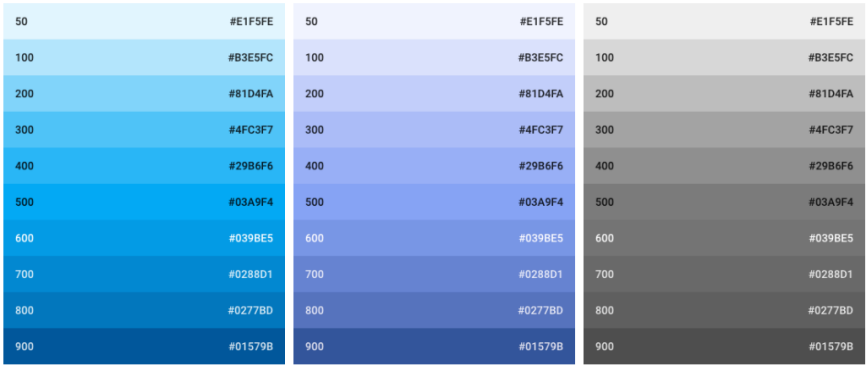


Imagen 1.23 Paletas de colores con diferente variación de brillo y contraste.

La segunda recomendación que nos da Zhang es buscar una paleta de colores que podamos encontrar fácilmente en la naturaleza. Por ejemplo, una progresión de color de amarillo a morado es una progresión natural de colores que podemos encontrar en un atardecer.



Imagen 1.24 Paleta de colores que puede ser localizada fácilmente en la naturaleza (atardecer)

La tercera y última recomendación que nos propone Zhang es elegir una gradiente de color, en lugar de elegir colores estáticos. Recomienda utilizar gradientes porque se puede realizar la variación de brillo y matiz de manera sencilla, además de que la progresión de colores se percibe de manera natural.

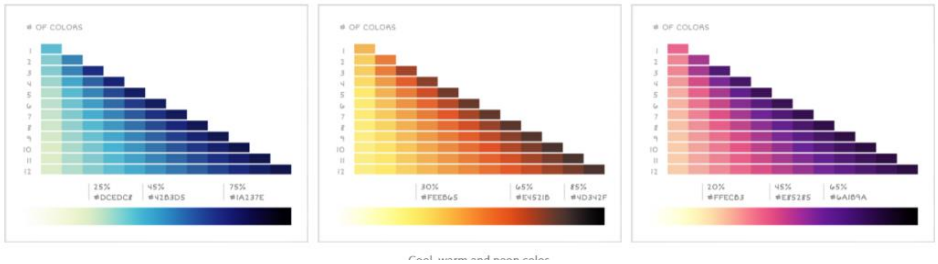


Imagen 1.25 Paletas de Colores propuestas por Samantha Zhang tomando en cuenta una variación amplia de brillo, contraste y matiz.

Zhang por último nos menciona que las paletas propuestas funcionarán de la manera esperada en visualizaciones de datos que contengan de 1 a 12 elementos.

Por su parte, Andy Kirk, en el libro Data Visualization a successful design (Kirk 92-106), nos menciona que la elección de la paleta de colores es un punto fundamental en la visualización de datos. Nos dice que si el color es usado incorrectamente puede que el usuario no comprenda correctamente lo que tratamos de transmitir o se distraiga con los colores mal usados.

Uno de los errores más comunes al elegir los colores para la visualización de datos, según Kirk, es elegir diferentes colores para representar información.

En el mapa que se muestra abajo, Kirk hace una comparativa visualizando el mismo mapa, pero con diferentes paletas de colores. En el primer caso,

el mapa no puede ser comprendido rápidamente porque los colores no nos indican de manera eficiente lo que quieren transmitir. En el segundo caso, el mapa puede ser comprendido rápidamente debido a que la selección de los colores se basa en variaciones del brillo y del matiz, siendo la información más representativa la del color más oscuro y la información menos representativa la del color más claro.

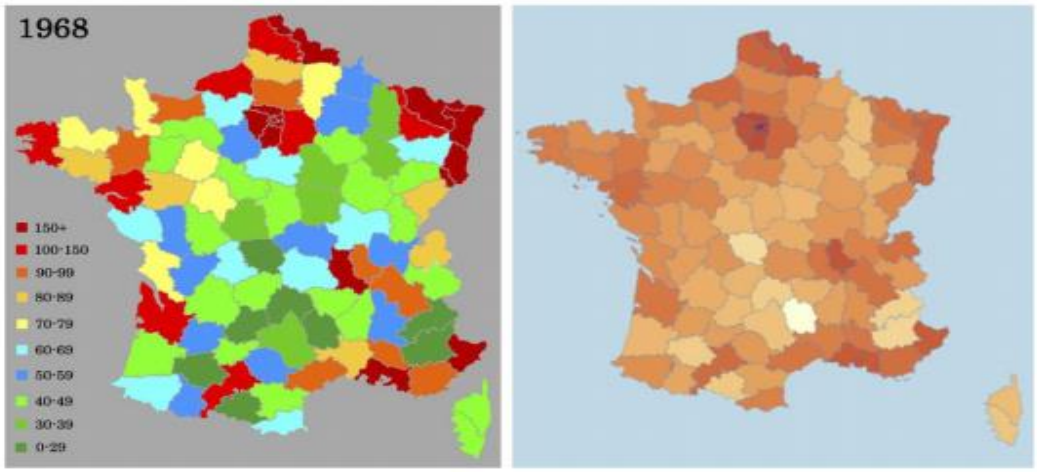


Imagen 1.26 Ejemplo de la Andy Kirk utilizando un mapa coroplético. En el primer mapa se utilizan diferentes colores para transmitir información. En segundo un solo color con un rango diferente de brillo, contraste y matiz.

Kirk también resalta que es importante poner atención en las personas daltónicas, y nos dice que al realizar las visualizaciones debemos elegir colores que ellos puedan percibir pensando también en la accesibilidad de las visualizaciones de datos. Una herramienta recomendada por Kirk es Vischeck[13]. Ésta herramienta es una buena opción para visualizar como ve los colores una persona daltónica y con base en ello, se facilita la elección de una paleta de colores accesible.

[13] http://www.vischeck.com/daltonize/

Un punto importante a tomar en cuenta cuando se va elegir color es que el ojo humano solo puede distinguir correctamente un máximo de doce colores en una clasificación de color[14]. Si se cuenta con más de doce colores en la clasificación probablemente sea conveniente encontrar una manera diferente de separar los datos.

Cuando se trata de utilizar colores que no sean una gradiente de color se debe tener en cuenta que los colores en algunos casos pueden crear una sensación de jerarquía.

Por último, en la guía de Amy Cesal, mencionada en el apartado anterior de la selección de fuentes. Cesal también contribuye en la recomendación de paletas de colores para la visualización de datos.

En los colores que nos propone Amy Cesal entran también las gradientes de colores para representar información, con variantes de brillo y de matiz. En la parte posterior incluye un apartado donde nos recomienda colores para casos específicos como la representación de información relacionada con política. Los colores que propone Amy Cesal para la política buscan representar solo dos vertientes, los partidos democráticos y los republicanos.

La propuesta de Amy Cesal también incluye colores para casos donde se pretenda resaltar cosas positivas o negativas o en caso de ser necesario un color para representar cuestiones monetarias.

[14] Visual Thinking by Design, Colin Ware.

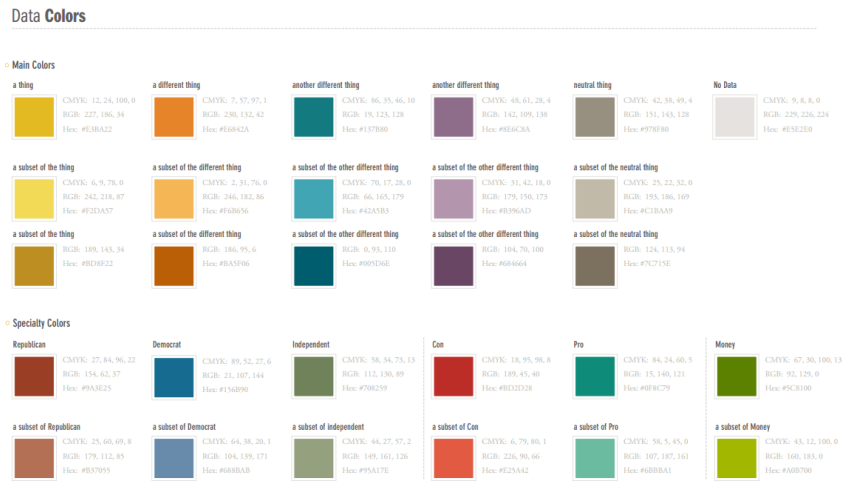


Imagen 1.27 La paleta de colores propuesta por Amy Cesal en su guía Data Visualization Style Guidelines URL

A continuación se muestra una tabla comparativa que junta las propuestas de los tres autores descritos anteriormente.

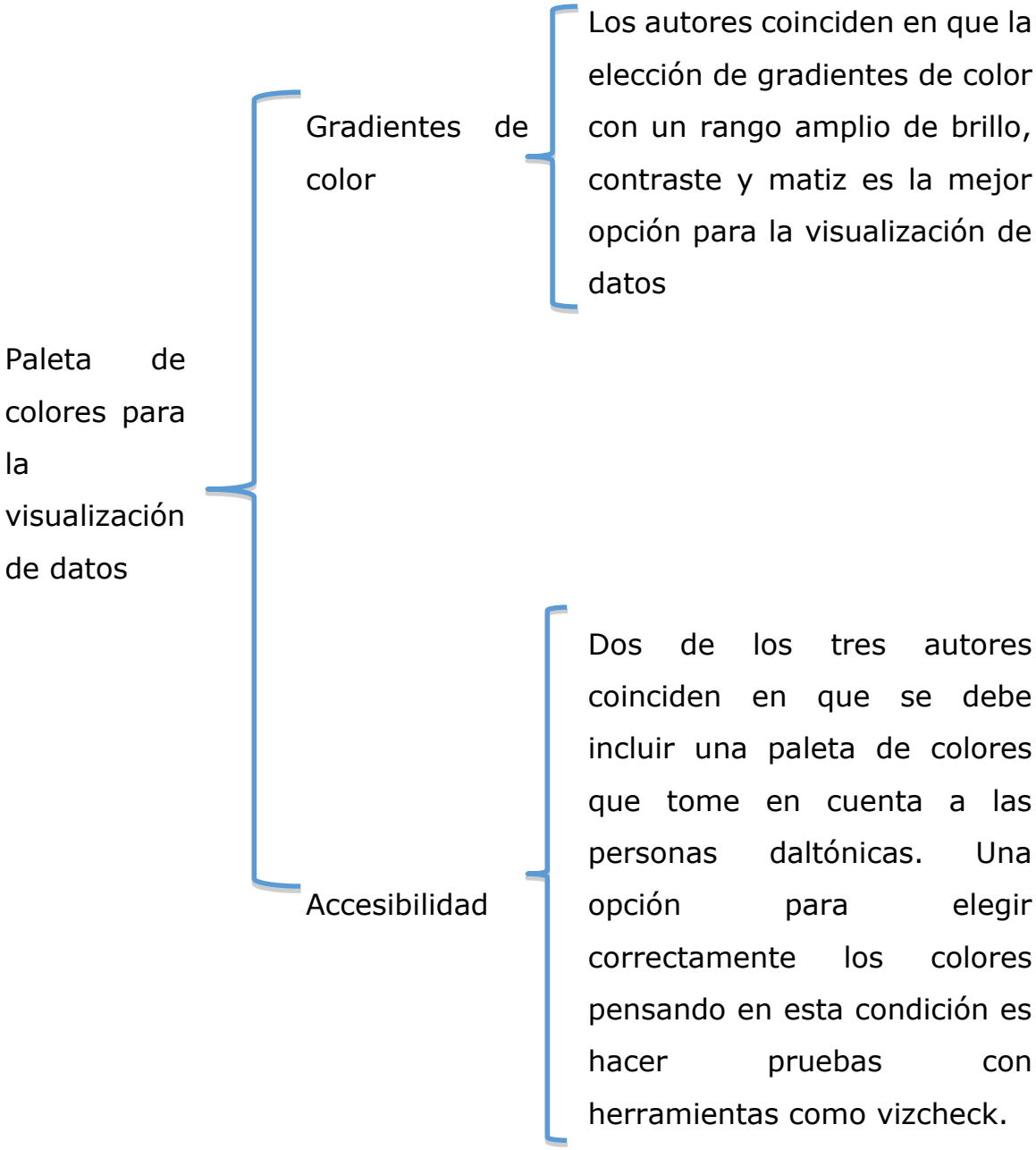
Características				
Autor	Accesibilidad para personas daltónicas	Paletas con gradientes de Color	Paletas con Colores de la Naturaleza	Paletas de colores para casos específicos
Samantha Zhang	X	*X*	*X*	
Andy Kirk	*X*	*X*		
Amy Cesal		*X*		*X*

Tabla 3. Tabla comparativa de las recomendaciones que nos proponen los 3 autores con respecto a la elección de la paleta de color

Una vez analizado las propuestas de los tres autores puedo concluir que los tres coinciden en que las gradientes de color, con las debidas

variaciones de brillo y contraste, son la mejor opción para la elección de la paleta de color en la visualización de datos. La accesibilidad, mencionado por dos de los tres autores, también jugó en papel importante, ya que se estima que el 8%[15] de la población mundial es daltónica. Hacer visualizaciones de datos tomando en cuenta esa condición es una medida necesaria e inclusiva.

[15] http://digital.csic.es/bitstream/10261/96086/1/Daltonismo_Menendez.pdf



Esquema 3. Esquema contiene la síntesis para la elección de la paleta de colores para la visualización de datos.

Conclusiones

Como conclusión se puede decir que los principales criterios a tomar en cuenta al momento de seleccionar un tipo de gráfico o mapa son los de comparación, relación y composición/parte de un todo. El criterio de distribución también es ampliamente considerado. También se identificó que existen muchos tipos de gráficos o mapas que pueden servir para más de una función. Una recomendación es realizar diferentes pruebas con varios tipos de gráficos para tener un panorama más amplio de cuál es el que se adapta mejor a las necesidades del proyecto.

Por su parte para la elección de tipografías para la visualización de datos se identificaron cuatro elementos a considerar. El primero de ellos es que se deben utilizar diferentes tamaños para los elementos de la visualización por ejemplo un tamaño para los títulos, otro para la información de los ejes, etc. El segundo elemento tiene relación y es que esos tamaños deben ser consistentes. El tercer elemento es que se debe evitar utilizar tipografías poco comunes o que puedan robar la atención del lector. Por último, las tipografías que se recomiendan utilizar son: Verdana, Arial, Helvética, Itc Franklin Gothic, y Gill Sans. Esto debido a que esas tipografías son de la familia Sans Serif y el lector común está acostumbrado a verlas y no roba la atención de la visualización.

Por último en la parte de la elección de paleta de colores se puede concluir que utilizar gradientes de color es la mejor opción, tomando en cuenta que dicha gradiente debe tener un rango amplio de brillo, contraste y matiz. Las paletas de colores que nos propone Zhang son un buen ejemplo de cómo utilizar esas variaciones de brillo y contraste. Como elemento final que tiene que ver con accesibilidad es recomendable revisar las paletas en programas como Vischeck, esto con el objetivo de inclusión a las personas daltónicas,

Con la anterior se puede decir que la hipótesis de comprobó parcialmente porque los criterios para seleccionar los gráficos sí coincidieron con los planteados en la hipótesis. Sin embargo, en el punto de selección de la tipografía la recomendación fue utilizar fuentes de la familia san serif, como Verdana, Arial, Helvetica, etc. Estás tipografías en general pueden cargar en cualquier navegador. Por último en la parte de la elección de paleta de colores también coincidió que los gradientes de color es la mejor opción al momento de elegir la paleta, sin embargo un punto que no se había tomado en cuenta en la hipótesis es la accesibilidad para las personas daltónicas, por lo que al momento de elegir la paleta se recomienda realizar pruebas con programas como Vischeck para garantizar la accesibilidad.

Capítulo 2 Elección de una librería JavaScript para la visualización de datos

Una vez que se ha seleccionada el tipo de gráfico, tipografía y paleta de colores para crear nuestra visualización de datos el próximo paso es identificar que herramienta se puede utilizar para lograr este objetivo.

Existen cientos de herramientas que podemos encontrar para la visualización de datos. Cada herramienta tiene características, ventajas y desventajas diferentes. Para está investigación se decidió analizar herramientas que se basan en el lenguaje de programación JavaScript.

Se eligió el lenguaje de programación JavaScript porque es uno de los lenguajes de programación más utilizados en la Web, además de que cuenta con un gran número de librerías y herramientas que podemos utilizar para la visualización de datos.

El objetivo del capítulo dos es identificar las librerías JavaScript para la visualización de datos que nos pueden ser útiles dependiendo de nuestro perfil, y nuestros conocimientos previos.

Este capítulo parte de la pregunta ¿ a partir del perfil, diseñador

Web o desarrollador Web, qué características deben poseer las librerías JS para que puedan ayudar a la visualización de datos? Y tiene como hipótesis que Las librerías JS, pensando en un perfil de diseñador Web, tienen como característica principal ser más visuales en el proceso de desarrollar las gráficas, además de que tienen un costo. Por otro lado las librerías JS, pensando en un perfil de desarrollador, tienen como característica principal basarse en usar solo código, además de ser de librerías de código abierto.

Para ello se seleccionan seis librerías JavaScript (estas librerías fueron seleccionadas mediante un comparativo de las librerías JS más utilizadas

para la visualización de datos), tres pensando en un perfil de diseñador Web y tres pensando en un perfil de desarrollador Web, donde se analizan algunas de las características más representativas de dichas librerías como son: costo, curva de aprendizaje para aprender a utilizarlas, cantidad de gráficos que ofrece cada librería, si cuentan con alguna comunidad o repositorio a la que podamos acudir si tenemos algún problema, si da soporte, sin son *responsive*, entre otras.

Capítulo 2.1 Librerías JavaScript pensando en un perfil de diseñador Web

Las librerías JavaScript que se han elegido tienen como característica principal que se puede realizar toda la visualización de datos mediante una interfaz, y casi sin utilizar código. Mediante está interfaz se pueden realizar todo el gráfico y posteriormente generará el código que podremos copiar y pegar dentro de la página donde lo requiramos.

Cuando hablamos de un perfil de diseñador Web, se piensa que el diseñador se siente más cómodo trabajando con herramientas más visuales y que requieran menos código JavaScript, sin embargo esto no los excluye de poder realizar las visualizaciones solo con código.

Los elementos que se analizan en el comparativo son los costos para utilizar estas herramientas, que tipo de interfaz poseen, si se puede importar datos de un tercero, por ejemplo, hojas de cálculo en Excel, archivos CSV, etc.

Otro elemento que se analiza es la cantidad de gráficos que tienen disponibles para que nosotros los podamos utilizar, las tipografías disponibles que nos ofrecen, y que, si la curva de aprendizaje para utilizarlas es alta o baja, si cuentan con soporte y si hay alguna serie de tutoriales o repositorios dónde podamos consultar cómo utilizarlas. Los últimos puntos que se toman en cuenta de estas herramientas son la compatibilidad con los navegadores, y si los gráficos que generan son *responsive* o no.

Las herramientas que se seleccionaron para hacer el estudio comparativo fueron: HighCharts, AM Charts y Live Gap Charts. Estas herramientas fueron seleccionadas mediante un comparativo de las librerías librerías JS

más utilizadas para la visualización de datos. Después de seleccionar las herramientas, se buscaron tres librerías que ofrecieran alguna interfaz de usuario y que no requirieran el dominio de un lenguaje de programación.

Capítulo 2.1.1 Librería High Charts Js

High Charts es una librería JavaScript que nos ofrece una forma rápida de añadir gráficos interactivos a las páginas Web. High Charts ofrece diversos paquetes para utilizar su librería. Una de sus opciones, que es la que se analiza en esta investigación, es la opción de poder realizar los gráficos online mediante una interfaz. Esta opción de High Charts[16], cuenta con tres planes para poder utilizarlo.



Imagen 2.1 Lista de precios de la Licencia de HighCharts URL

El primer plan es gratuito, con el inconveniente de que el logotipo de High Charts estará presente en todas las visualizaciones que se realicen. El segundo plan cuesta 15 dólares al mes, pero nos permitirá crear las visualizaciones sin el logotipo de Hich Charts. El tercer plan nos permitirá tener acceso al código fuente, además de poder importar la información para la visualización mediante Google SpreadSheet.

[16] https://cloud.highcharts.com/

La herramienta online de High Charts funciona al importar datos en formato CSV[17] o simplemente llenando los datos en la misma herramienta mediante tablas y columnas.

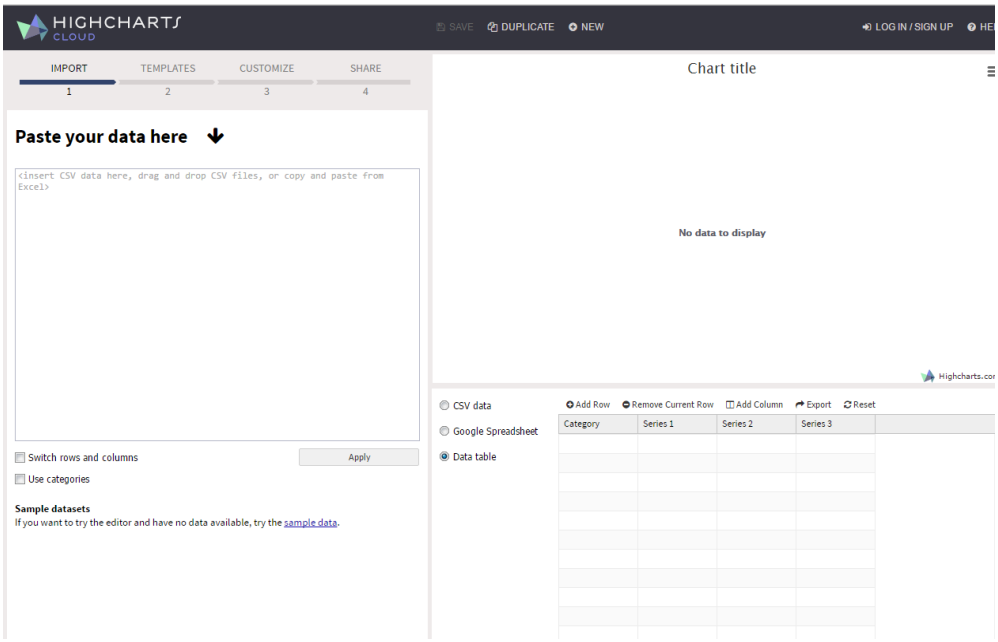


Imagen 2.2 Interfaz para crear gráficos de HighCharts, llamada Cloud High Charts URL

Una vez que se hayan ingresado o importado los datos High Charts nos ofrece once diferentes tipos de gráficos que podemos utilizar para crear las visualizaciones de datos. Dentro de los gráficos que nos permite usar Highcharts se encuentran: gráficos de área, de columna, de barras, de dispersión, de línea, de burbujas, de pie, polar[18], de área y columna apiladas, y de caja y bigote.

[17] Comma Separate Values

[18] El gráfico polar es el mismo gráfico de araña que se describió en el capítulo 1.

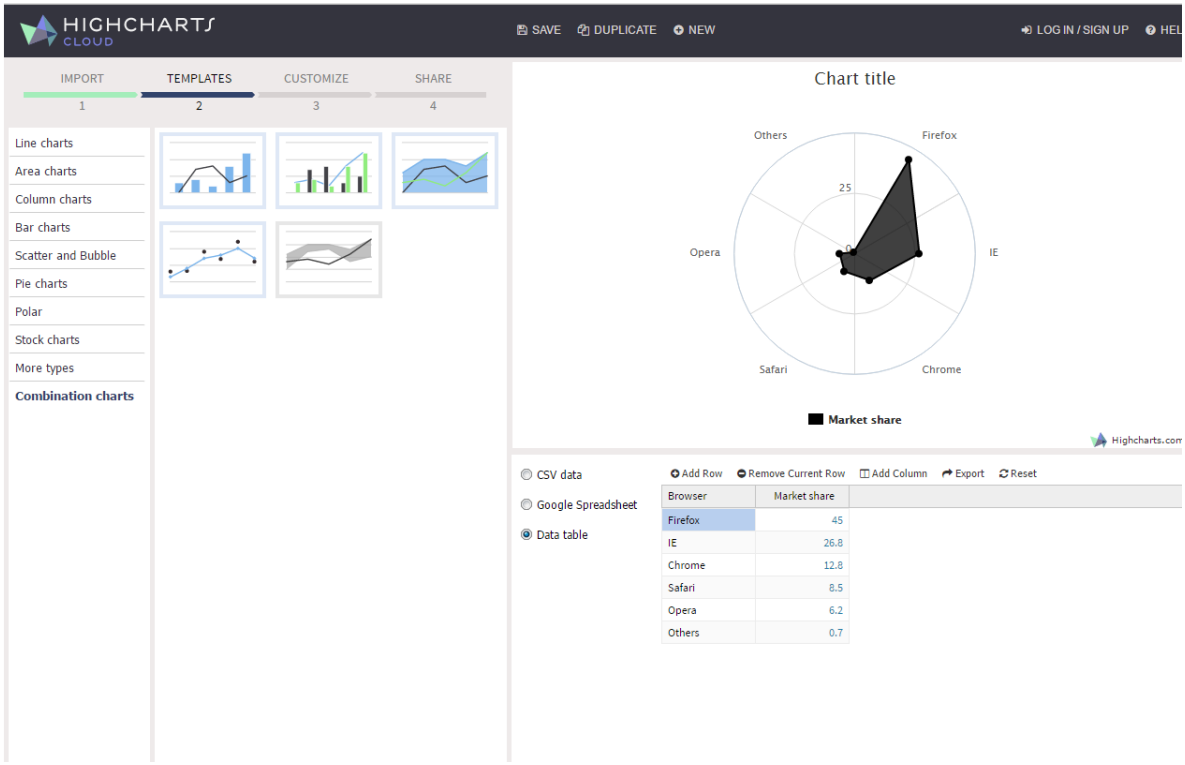


Imagen 2.3 Proceso para crear las primeras visualizaciones de datos con el cloud de HighCharts. URL

Una vez seleccionado el tipo de gráfico se procede a la parte de edición donde podremos modificar los colores y las tipografías.

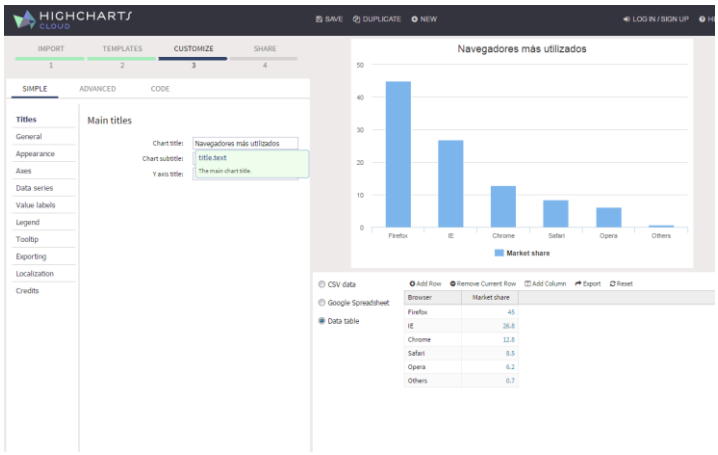


Imagen 2.4 Segundo paso para crear visualización de datos que consiste en seleccionar los colores, tipografías, etc. URL

Dentro de la interfaz de edición se puede seleccionar la tipografía que queremos usar. High Charts da la opción de elegir entre 13 tipografías distintas. Además de la tipografía también se podrán seleccionar los colores que queramos, así como el tamaño de las fuentes para los títulos, los valores de los ejes, etc.

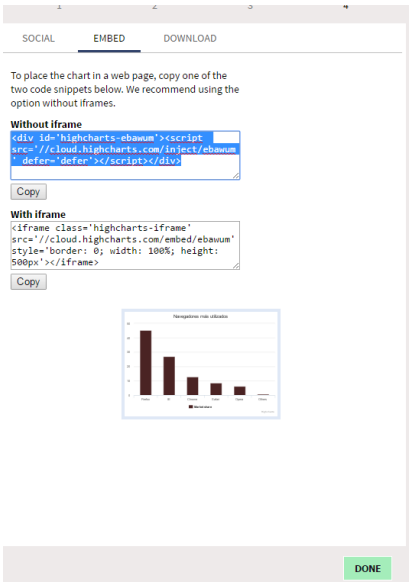


Imagen 2.5 Último paso para concluir la visualización el cual consiste en exportar el código JavaScript. URL

Una vez que se haya concluido el gráfico, basta con incluir el código que nos genera la página y agregarlo a nuestra página Web.

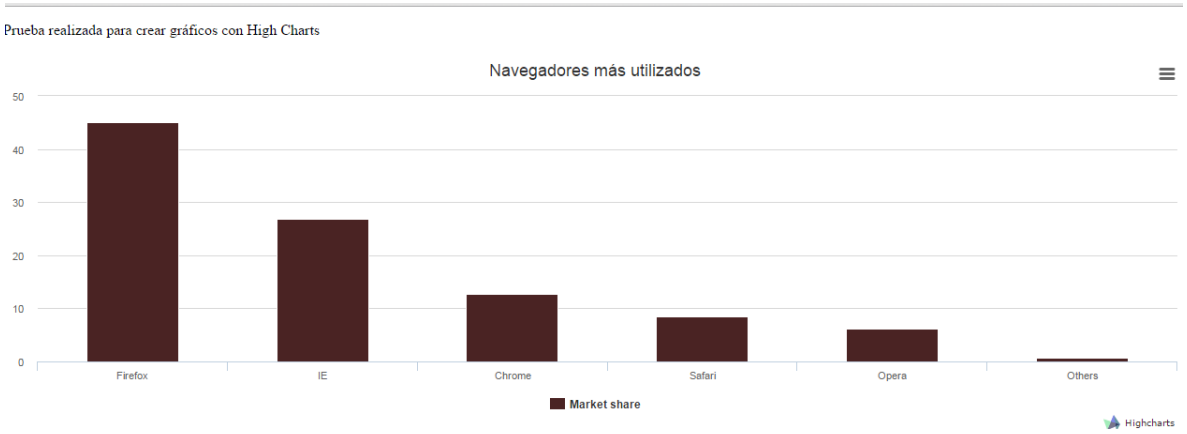


Imagen 2.6 Ejemplo de un gráfico generado mediante el Cloud de HighCharts. URL

High Charts cuenta con una amplia documentación sobre cómo utilizar la librería en caso de que queramos realizar todo el código nosotros mismos. Para su herramienta online cuenta con un breve video introductorio donde nos muestra que se puede realizar con esa herramienta, además los gráficos generados mediante esta librería son *responsive*. La herramienta online es intuitiva y fácil de utilizar.

High Charts también permite la creación de mapas interactivos mediante su versión HighMaps, sin embargo está opción solo se encuentra disponible para realizar el código nosotros mismos, no cuenta con ninguna interfaz como la analizada en este capítulo. High Charts ofrece soporte técnico a los usuarios de sus licencias.

2.1.2 Librería AM Charts

AM Charts es una librería JavaScript que ayuda crear visualizaciones de datos de manera sencilla. Esta librería no solo permite generar gráficos, sino también cuenta con la opción de generar mapas interactivos. La librería puede descargarse para utilizarse o cuenta con una opción que

nos ayuda a realizar los gráficos mediante un interfaz, está es la opción que se va a analizar.

Ésta librería cuenta con diversos métodos de pago de acuerdo a las necesidades de cada persona. En un principio la librería puede ser usada de manera gratuita, con el inconveniente de que aparecerá una leyenda en cada visualización que hagamos que dirá que fue realizada con AM Charts.

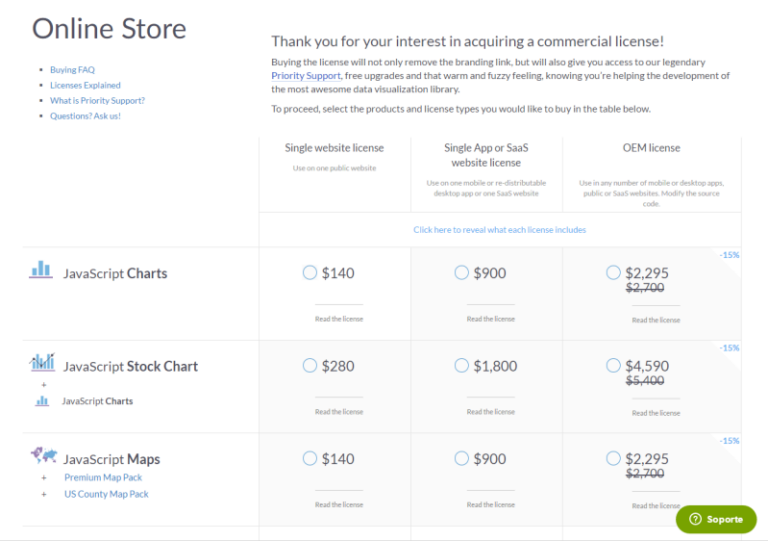


Imagen 2.7 Lista de precios de la licencia para utilizar AM Charts. URL

El primero método de pago que se menciona es para utilizarlo en un sitio Web simple. Hay tres opciones que se ofrecen, la primera es para utilizar solo los "JavaScript Charts", que nos da acceso a utilizar solo los gráficos de AM Charts. Esta opción tiene un costo de 140 dólares por la licencia. La segunda opción incluye los gráficos de "JavaScript Stock Charts" y los "JavaScript Charts", y tiene un costo de 280 dólares. La diferencia entre estos dos es que los JavaScript Stock Charts se utilizan principalmente

para mostrar estados financieros, aunque también pueden ser utilizados para realizar cualquier visualización de datos.

La tercera opción es la licencia para utilizar los mapas interactivos y tiene un costo de 140 dólares, al comprar esta licencia se obtienen algunos mapas interactivos extras que no están disponibles en la versión gratuita.

El siguiente plan es para utilizarlo en aplicaciones SaaS[19], y se ofrece utilizar los mismos paquetes pero con un costo mayor. Por ejemplo la licencia para utilizar solo los JavaScript Charts tiene un costo de 900 dólares, para utilizar los JavaScript Stock Charts y los JavaScript Charts es de 1800 dólares, y la licencia para utilizar los mapas interactivos es de 900 dólares.

El último plan que se menciona es para utilizarlos en cuantas páginas Web, o aplicaciones necesitemos y tienen un costo de 2,295 dólares para la licencia de los JavaScript Charts, 4,590 dólares para la licencia de los JavaScript Stock Charts y JavaScript Charts y 2,295 para la licencia de los JavaScript Maps.

El pago por cualquiera de las licencias anteriores te da acceso al soporte Premium que ofrece AM charts además de eliminar la etiqueta de la librería en las visualizaciones que se realicen.

AM Charts ofrece dos opciones para realizar gráficos online mediante una interfaz. La primera es mediante el live editor[20] de AM Charts, dónde podemos crear gráficos de columnas, de barras, de línea, de área, de pie o dona, de burbuja y de radar entre otros.

[19] Software as a Service por sus siglas en inglés.
[20] https://live.amcharts.com

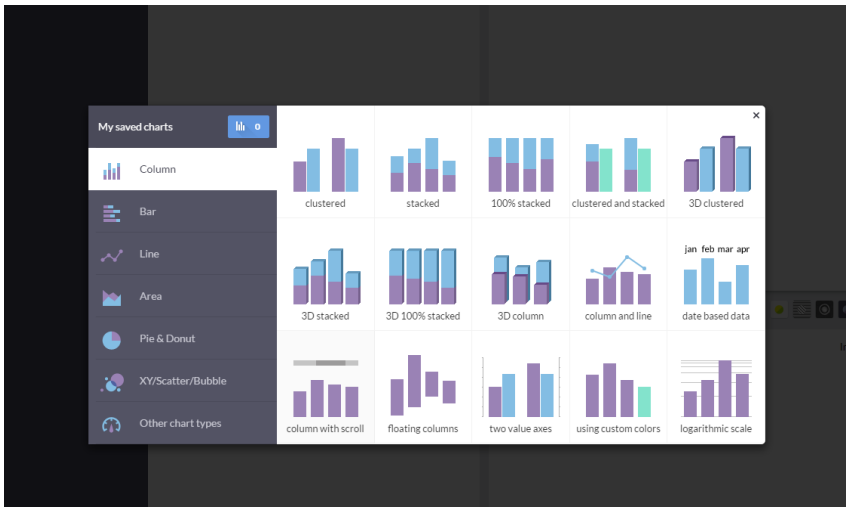


Imagen 2.8 Tipos de gráficos que ofrece AM Charts. URL

Mediante una interfaz similar a la de High Charts mencionada en el capítulo anterior es posible crear visualizaciones de datos con AM Charts.

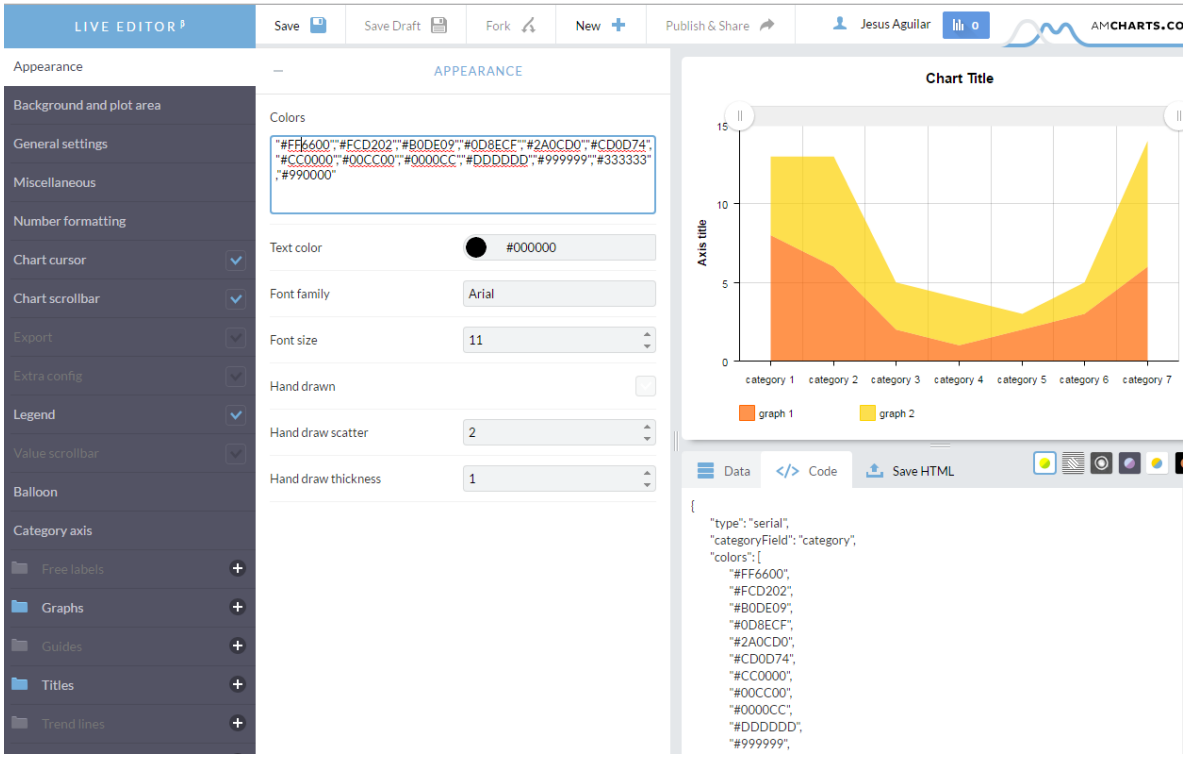


Imagen 2.9 Interfaz de AM Charts Live Editor para crear visualizaciones de datos. URL

En la parte inferior derecha podremos ingresar la información que vayamos a utilizar mediante filas y columnas. Además de incluir una opción donde se puede importar archivos en Excel y archivos en CSV.

En la parte de apariencia, nos permite modificar los colores mediante número hexadecimales, así como elegir la tipografía y el tamaño de la fuente. En las secciones posteriores se podrán modificar otras características, por ejemplo, si se desea que el gráfico aparezca con una animación, la duración de la animación, los valores de los ejes, la información que aparecerá en las etiquetas de los ejes, su tamaño, etc.

Una vez que se ha terminado de realizar el gráfico, basta con pulsar el botón que dice "Save HTML", está opción nos proporcionará todo el código generado por AM Charts y después deberemos pegarlo en la página que vamos a utilizar. Los gráficos generados por AM Charts con responsive.

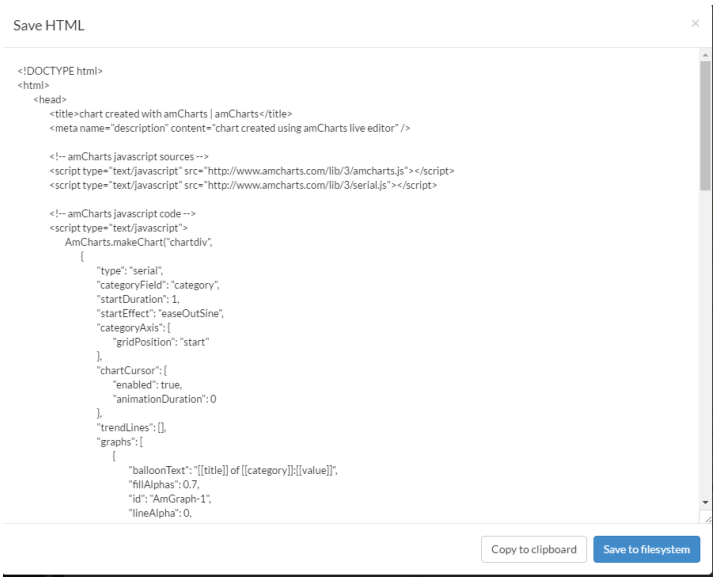


Imagen 2.10 Código JavaScript generado mediante AM Charts

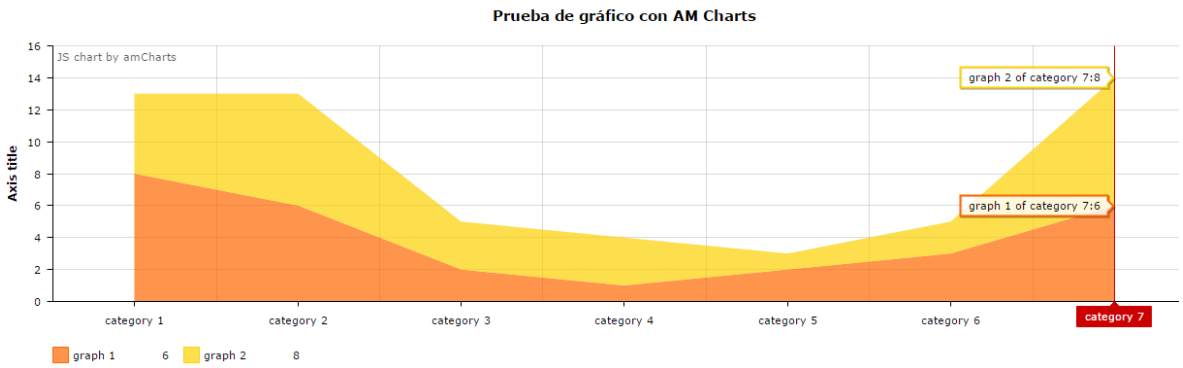


Imagen 2.11 Ejemplo de gráfico de área apilada realizado con AM Charts.

La segunda opción online que ofrece AM Charts es la de realizar mapas interactivos mediante una interfaz también. Esta opción se llama pixelmap[21]. Esta herramienta aún se encuentra en su etapa beta, sin embargo, nos permite realizar algunos mapas interesantes.

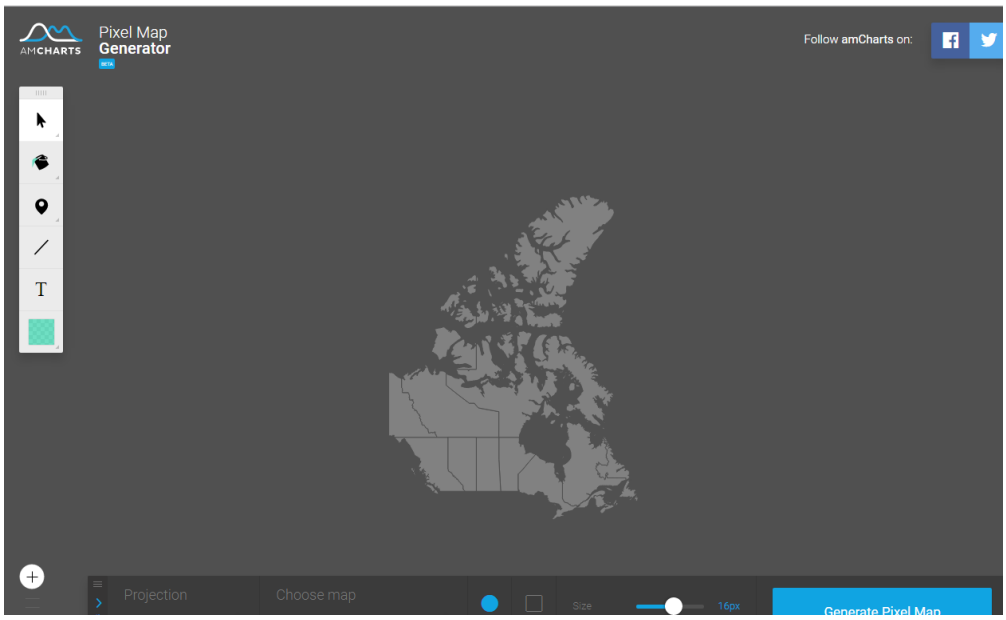


Imagen 2.12 Herramienta Pixel Map de Am Charts para crear mapas geográficos. URL

[21] http://pixelmap.amcharts.com/

En la parte inferior se puede seleccionar el mapa que se desee utilizar, AM Charts ofrece una gran cantidad de países con los cuales podemos trabajar. Una vez seleccionado el mapa que se va utilizar, en la parte superior izquierda hay una pequeña interfaz que podemos utilizar. Dentro de esta sección hay un bote de pintura donde se puede cambiar el color de algunas de las regiones del mapa. Además, podemos agregar texto para complementar la información.

Una vez finalizado la creación de los mapas, bastará con repetir el proceso de generar el código y posteriormente lo pegaremos en la página donde se desea que se visualice.

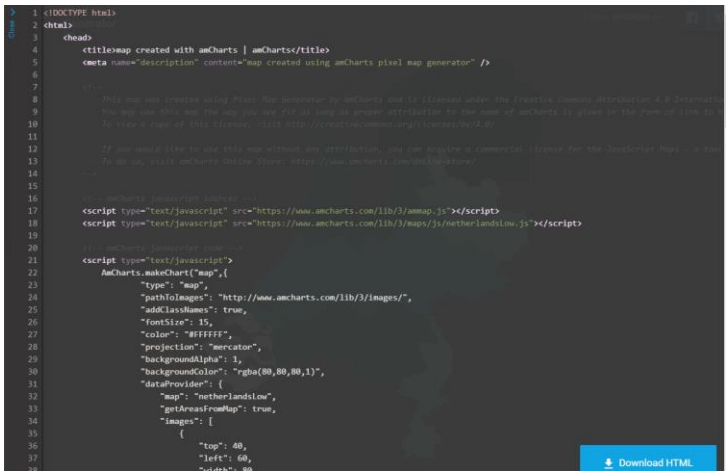


Imagen 2.13 Código JavaScript generado mediante la herramienta Pixel Map

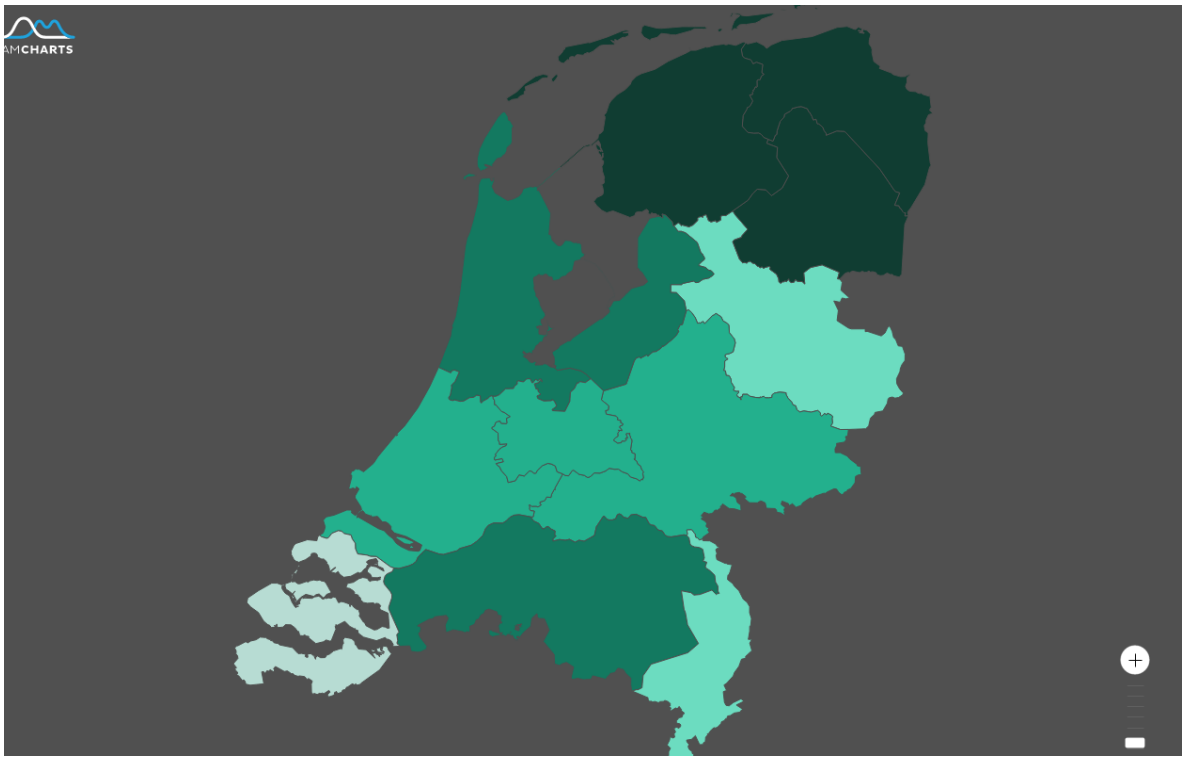


Imagen 2.14 Código Ejemplo de mapa geográfico generado con Pixel Map

AM Charts proporciona una serie de tutoriales en su página dónde se puede aprender cómo utilizar las librerías, en caso de querer realizar todo el código nosotros mismos. Una pequeña desventaja es que no ofrece tutoriales sobre cómo utilizar su herramienta online. Sin embargo, la interfaz es bastante intuitiva y fácil de utilizar.

En caso de comprar la licencia la librería proporciona soporte en caso de tener algún inconveniente.

2.1.3 Live Gap Charts

Live Gap Charts[22] es la tercera herramienta que se va a analizar en esta investigación. Live Gap Charts al igual que las herramientas anteriormente descritas utiliza una interfaz para que podemos crear visualizaciones de datos. A diferencia de High Charts o AM Charts, Live Gap Charts se enfoca solamente en la creación de gráficos, no cuenta con ninguna herramienta para crear mapas interactivos. Live Gap Charts es una opción completamente gratuita.

Esta herramienta tiene como opción la importación desde un archivo de Excel o una hoja de cálculo. Además de contar con 9 gráficos que podemos utilizar. Estos gráficos son: gráfico de línea, de área, de barra, de pie, de rosa, de dona, de radar o araña, de paso, y de área apilada.

La interfaz es parecida a las dos librerías que se muestran anteriormente. Desde la interfaz se puede modificar el color de los elementos, la tipografía, el tamaño de la fuente, si la visualización contará con una animación, etc.

Una desventaja de esta herramienta es que dentro de las tipografías que nos ofrecen ninguna de ellas cuenta con las que se recomiendan en el capítulo anterior.

[22] http://charts.livegap.com/

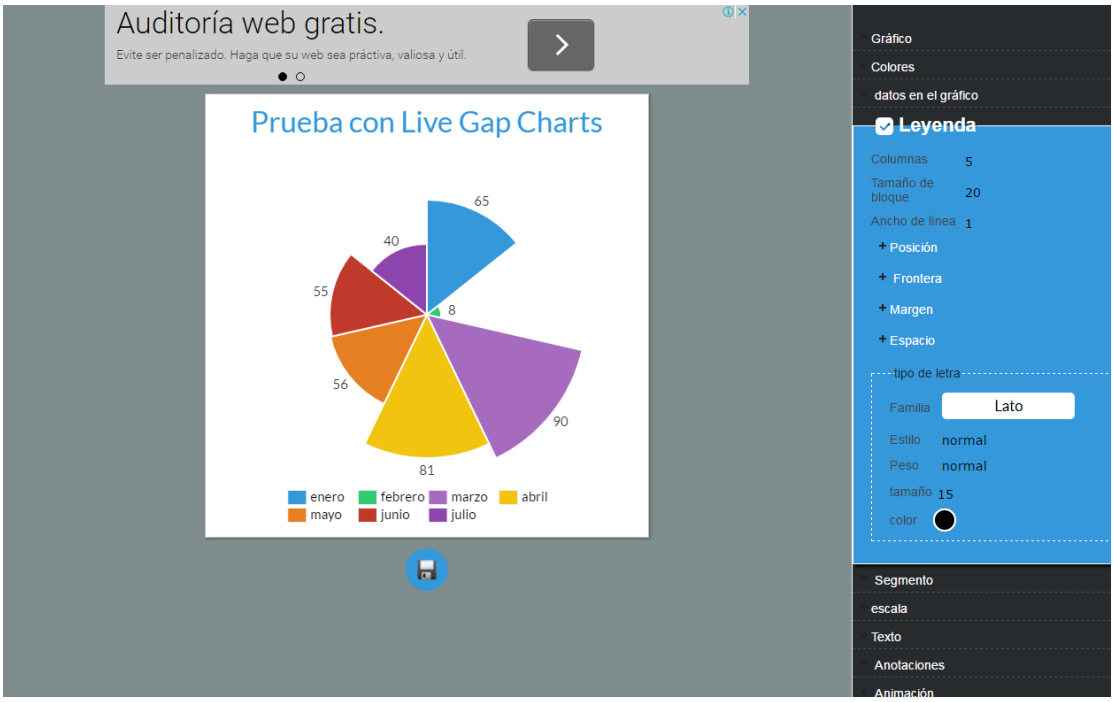


Imagen 2.15 Interfaz de la herramienta Live Gap Charts para crear visualizaciones de datos. URL

Una vez que hayamos terminado de realizar nuestra visualización bastará con copiar y pegar el código que genera la herramienta y pegarlo en nuestra página para poder verlo.

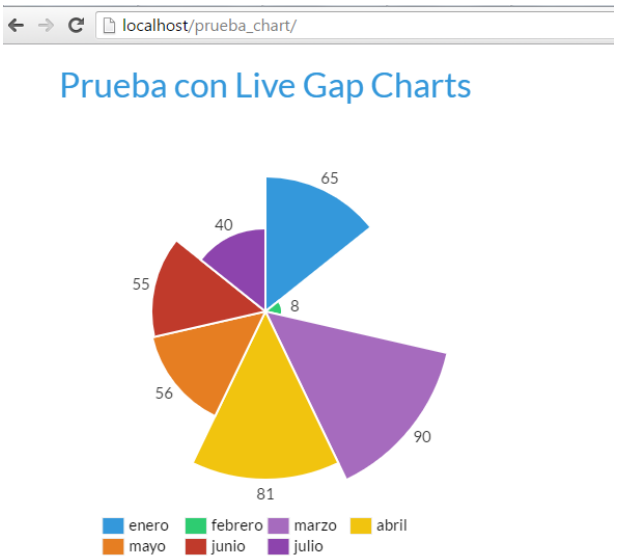


Imagen 2.16 Ejemplo de gráfico generado con Live Gap Charts

Algo a tomar en cuenta al momento de utilizar esta herramienta es que los gráficos que genera no son responsive, lo cual supone una gran desventaja en comparación con las librerías High Charts y AM Charts.

Otro punto a tomar en cuenta es que dentro de la página no se muestra documentación en caso de que surja algún problema. Sin embargo, la interfaz es intuitiva y fácil de utilizar.

Librería	High Charts(Cloud)	AM Charts (Live Editor)	Live Gap Charts
Costo	Gratis con marca de agua. Sin maraca 15 dólares al mes, o 30 el premium	Gratis con marca de agua	Gratis
Tutoriales/Repositorios	Sí	Sí	No
Responsive	Sí	Sí	No
Grado de dificultad para usarla	Intuitiva y fácil de usar	Intuitiva y fácil de usar	Intuitiva y fácil de usar
Cantidad de gráficos disponibles	11 diferentes tipos de gráficos, con algunas variaciones	7 diferentes tipos de gráficos con algunas variaciones	9 diferentes tipos de gráficos
Mapas Geográficos	No	Sí, pero solo mapas de calor o coropléticos	No
Soporte	Sí, pero en la versión de paga	Sí, pero en la versión de paga	No
Crossbrowser	Sí	Sí	No

Tabla 4. Tabla comparativa de las librerías pensando un perfil de diseñador Web.

Las tres librerías analizadas anteriormente tienen varias cosas en común. Por ejemplo, la interfaz que presentan para comenzar a crear la visualización de datos es similar y no se requiere una gran cantidad de esfuerzo para empezar a utilizarlas. Las tres son intuitivas y fáciles de utilizar. En cuestión de precios las tres cuentan con alguna opción que puede ser gratuita, aunque aparecerá una marca de agua. Por otro lado ofrecen una opción que nos permite usarla sin marca de agua y nos da soporte.

Las primeras dos cuentan con una gran ventaja la cual es que todos los gráficos que generan son responsive y se pueden visualizar en la mayoría de los navegadores.

Por último, en caso de querer realizar mapas geográficos, solo AM charts cuenta con una opción para poder realizarlos.

2.2 Librerías JavaScript pensando en un perfil de desarrollador Web

Las librerías JavaScript que fueron elegidas pensando en un perfil de desarrollador Web tienen como característica que es necesario escribir el código nosotros mismos, y que algunas con de código abierto. Se eligieron pensando que un desarrollador Web tiene una base más sólida en algún lenguaje de programación que un diseñador Web y que la curva de aprendizaje para poder usarlas no será tan alta.

Estas librerías también tienen como característica que son más personalizables, ya que al ser de código abierto se pueden probar varias posibilidades al momento de realizar visualizaciones.

Las características que se analizan en el comparativo son: costo de las librerías, cantidad de gráficos están disponibles, curva de aprendizaje

para utilizarlas, si cuentan con algún repositorio o tutoriales para empezar a utilizarlas, si los gráficos que generan son responsive, compatibilidad con la mayoría de los navegadores.

Las librerías que fueron elegidas son: D3.js, AnyChart.js y Chart.js.

2.2.1 Librería D3.js

D3.js es una de las librerías JavaScript más populares para crear visualizaciones de datos. D3, se basa principalmente en el uso de SVG, HTML Y CSS para realizar dichas visualizaciones. D3 también permite manipular cualquier parte de DOM[23].

Actualmente muchas otras librerías JavaScript se basan en D3 para crear visualizaciones de datos.

Una de las características principales de D3 es su amplia documentación[24] y tutoriales[25] disponibles en su página Web. Además, por ser una librería tan popular es posible encontrar una gran serie de recursos disponibles en la red. Sin embargo no cuenta con ningún soporte en caso de tener algún problema al utilizarla, por otro lado en algunos foros como Stack Overflow[26] o gitter[27], las comunidades que resuelven dudas relacionadas con d3 regularmente están muy activas.

Esta librería no tiene costo alguno, y al ser de código abierto solo basta con descargarla, o utilizar un CDN para poder utilizarla en algún proyecto.

D3 además tiene la ventaja de poder utilizar mapas geográficos para nuestras visualizaciones de datos.

---

[23] Document Object Model
[24] https://github.com/d3/d3/wiki
[25] https://github.com/d3/d3/wiki/Tutorials
[26] http://stackoverflow.com/questions/tagged/d3.js
[27] https://gitter.im/d3/d3

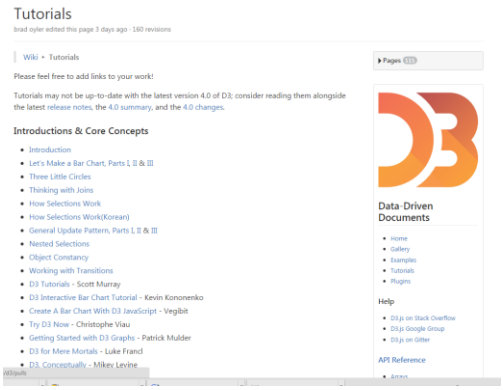


Imagen 2.17 Apartado de tutoriales disponibles en D3.js URL

D3 ofrece una galería de ejemplos de la mayoría de las visualizaciones que podemos crear con ellos.



Imagen 2.18 Galería de ejemplos que tiene D3.js para consultar diferentes tipos de gráficos. URL

Dentro de cada ejemplo de su galería se proporciona el código HTML, CSS y JavaScript que fue utilizado para crear la visualización.

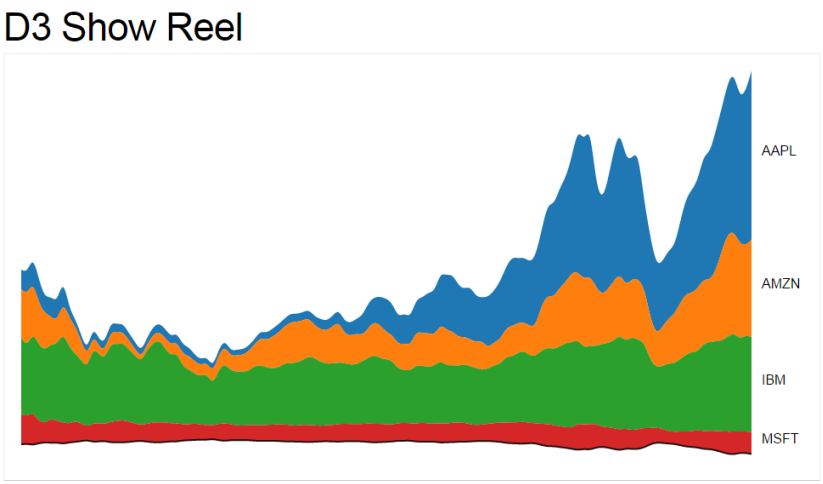
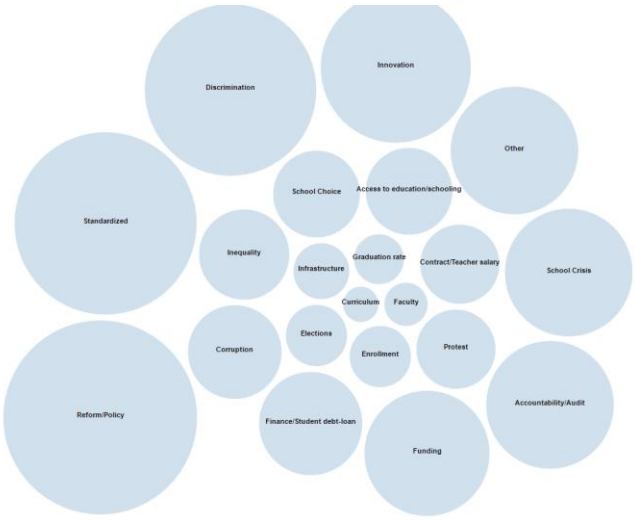


Imagen 2.19 Ejemplo de gráfico expuesto en la galería de D3 URL

Una de las desventajas que puede llegar a tener D3 es que requiere una buena base de conocimiento en JavaScript, además de que la curva de aprendizaje es amplia. Para poder utilizar correctamente D3.js se necesita invertir tiempo en familiarizarse con la sintaxis.

En la parte posterior se muestra un ejemplo de gráfico de burbujas generado por D3.js y en la parte posterior se muestra el código JavaScript que fue utilizado para crear dicho gráfico.



```
function burbujas(){
        var width = 1200;
                var height = 1200;

                var canvas = d3.select("body").append("svg")
                                        .attr("width", width)
                                        .attr("height", height)
                                    .append("g")
                                    .attr("transform", "translate(50, 50)");

                var pack = d3.layout.pack()
                            .size([width, height - 50])
                            .padding(10);

                d3.json("states.json", function(data) {
                        var nodes = pack.nodes(data)

                                var node = canvas.selectAll(".node")
                                                        .data(nodes)
                                                        .enter()
                                                        .append("g")
                                                        .attr("class", "node")
                                                        .attr("transform", function(d) {return "translate(" + d.x + "," + d.y + ")";})

                                node.append("circle")
                                        .attr("r", function (d) { return d.r; })
                                        .attr("fill", function(d){ return d.children ? "#fff" : "steelblue"; })
                                        .attr("opacity", 0.25)
                                        .attr("stroke", function(d) {return d.children ? "#fff" : "ADADAD"})
                                        .attr("stroke-width", 2);

                                node.append("text")
                                        .text(function(d) { return d.children ? "" : d.name;})
                                        .attr("text-anchor", "middle")
                                        .attr("font-weight", "bold")
                                        .attr("font-size", "11px");

                })
}

window.addEventListener("load", burbujas());
```

Imagen 2.20 Ejemplo de gráfico de burbuja generado con la librería D3.js

D3 también ofrece la posibilidad de utilizar mapas geográficos para la creación de visualizaciones de datos.

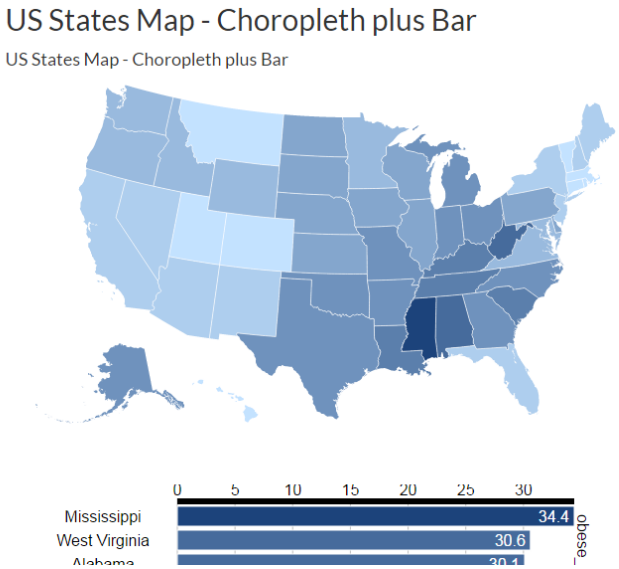


Imagen 2.21 Ejemplo de mapa geográfico generado con la librería D3.js URL

2.2.2 Librería AnyChart.js

AnyChart.js es una librería JavaScript basada en HTML5. Esta librería permite crear gráficos interactivos de manera sencilla. Esta librería es cross-browser, y funciona correctamente en muchos dispositivos y navegadores.

AnyChart[30] cuenta una versión gratuita para descargar y utilizarla si te registras en su sitio Web. También ofrece opciones de pago. La ventaja de las opciones de pago es que ofrecen soporte en caso de cualquier complicación. Las opciones de pago van desde los 79 dólares por un sitio Web, hasta los 1,499 dólares para que la licencia se pueda usar por todo un equipo de desarrolladores. En caso de requerir soporte y no tengamos una licencia, el sitio Web de AnyChart nos sugiere utilizar el apartado de StackOverflow[31]. En este apartado la gente suele consultar algunas dudas, este apartado es una buena opción en caso de tener problemas.

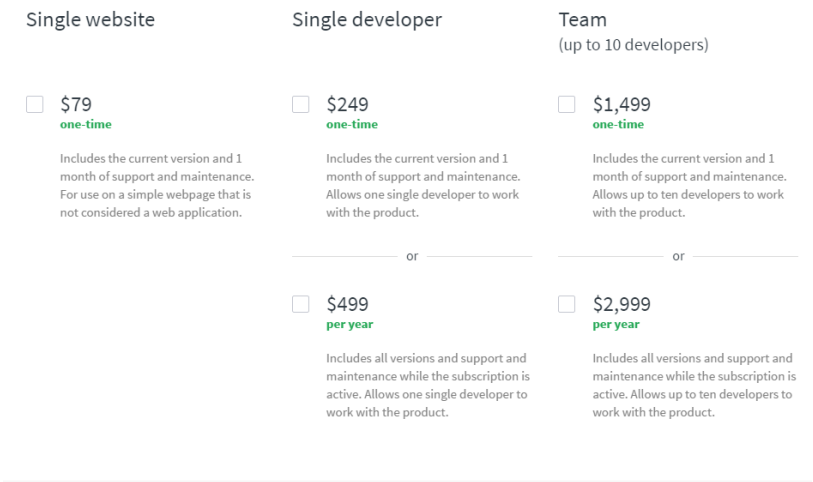


Imagen 2.22 Lista de precios de licencia para utilizar la librería AnyChart.js URL

AnyChart ofrece una opción llamada Playground, donde se pueden consultar todos los gráficos que pueden ser realizados con esta

[30] http://www.anychart.com/
[31] http://stackoverflow.com/questions/tagged/anychart?page=1&sort=newest&pagesize=15

herramienta. Además, podemos dar un pequeño vistazo al código JavaScript y a la sintaxis que se utiliza para darnos una idea de cómo funciona.

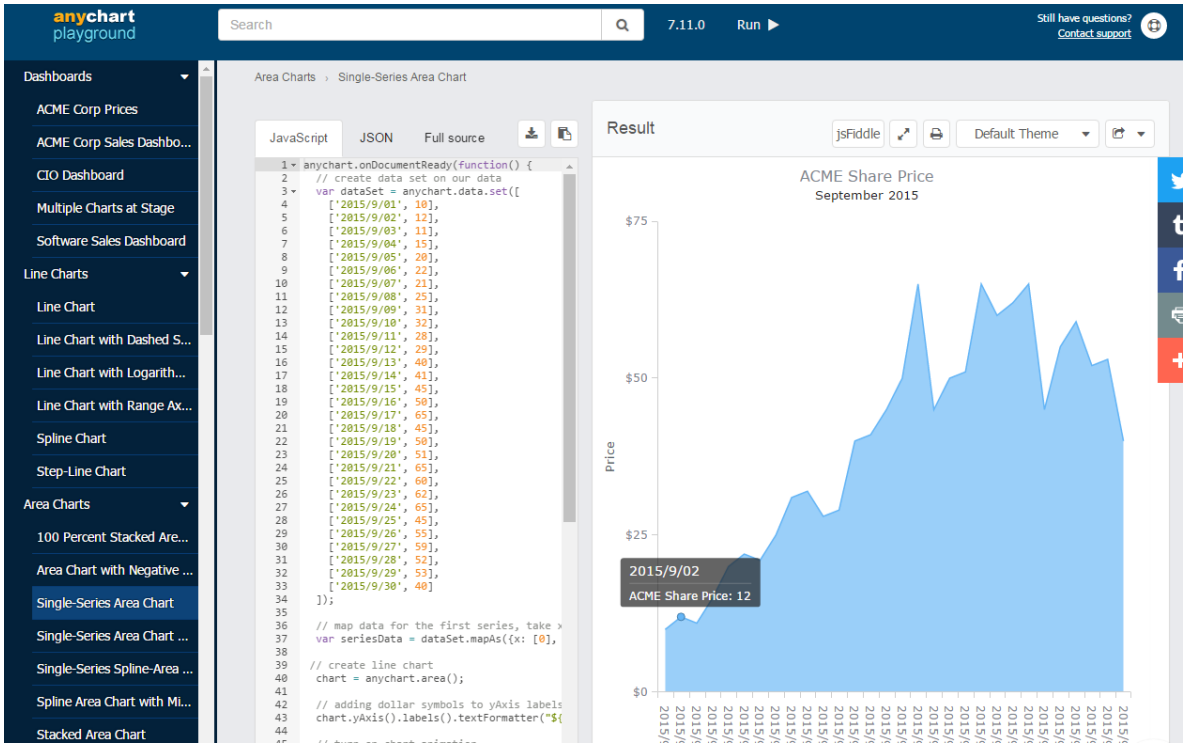


Imagen 2.23 Playground de la librería AnyChart.js URL

AnyChart ofrece hasta 36 diferentes tipos de gráficos que podemos utilizar para la creación de visualización de datos.

Any charts ofrece una documentación amplia sobre cómo empezar a utilizar esta librería. Además ofrece ejemplos bien detallados sobre cómo realizar cada tipo de gráfico que ofrecen.

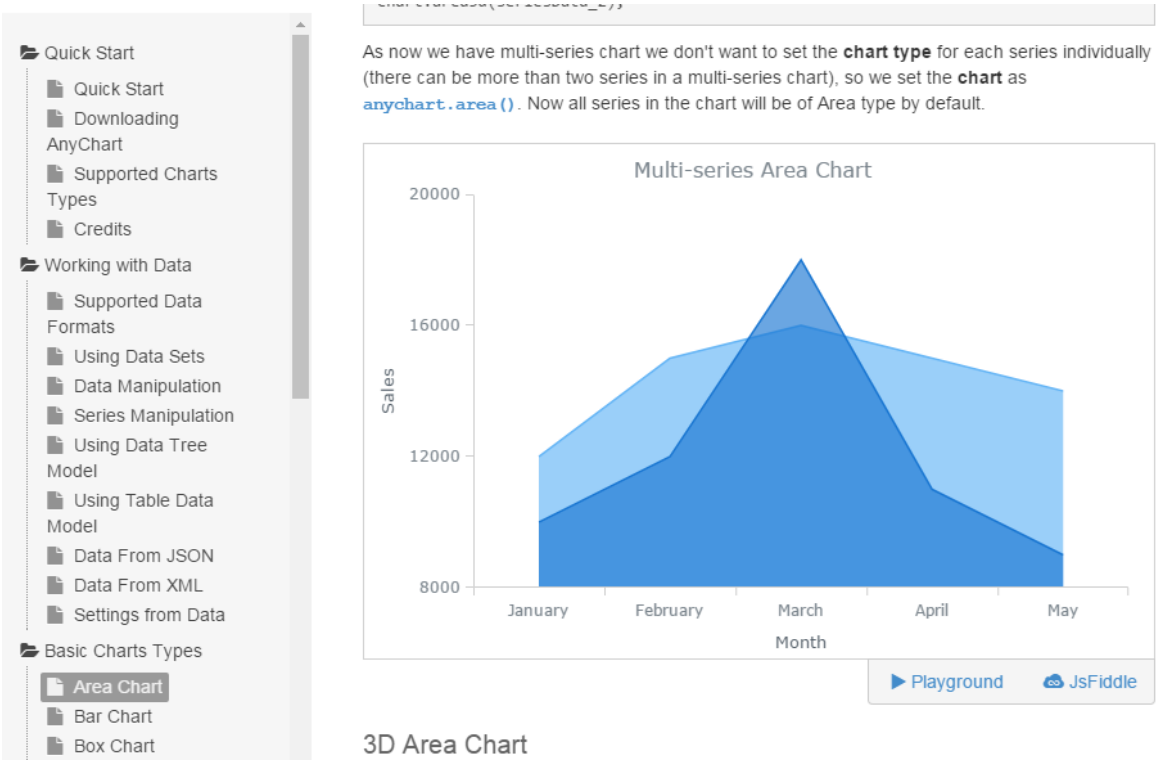


Imagen 2.24 Ejemplo de gráfico de área apilada realizado con AnyChart.js URL

En la parte posterior se muestra el código utilizado para crear el gráfico anterior. Como se puede observar con pocas líneas de código se puede crear un gráfico interesante con AnyChart.js.



```
anychart.onDocumentReady(function() {

  // data
  var dataSet = anychart.data.set([
    ["January", 12000, 10000],
    ["February", 15000, 12000],
    ["March", 16000, 18000],
    ["April", 15000, 11000],
    ["May", 14000, 9000]
  ]);

  // map data for the each series
  var seriesData_1 = dataSet.mapAs({x: [0], value: [1]});
  var seriesData_2 = dataSet.mapAs({x: [0], value: [2]});

  // chart type
  chart = anychart.area();

  // set title
  chart.title("Multi-series Area Chart");

  // set axes titles
  var xAxis = chart.xAxis();
  xAxis.title("Month");
  var yAxis = chart.yAxis();
  yAxis.title("Sales");

  // using data in series
  var series1 = chart.area(seriesData_1);
  series1.name("2004");
  var series2 = chart.area(seriesData_2);
  series2.name("2005");

  // draw chart
  chart.container("container");
  chart.draw();
});
```

Imagen 2.25 Código JavaScript que se utilizó para crear el gráfico de área anterior

El tipo de datos con los que podemos trabajar en AnyChart.js son: JavaScript, CSV, XML y JSON. Trabajar con este tipo de datos es una gran ventaja debido a que la sintaxis es sencilla y nos podemos adaptar rápidamente. Una característica más de esta librería es que todos los gráficos que realizamos en ella son responsive.

Anychart cuenta también con una librería llamada AnyMap[33], la cual permite la creación de mapas geográficos de manera sencilla. Para utilizarla solo basta con descargarla y comenzar a utilizarla. Además de contar con una documentación completa para utilizar los diferentes tipos de mapas geográficos.

[33] http://www.anychart.com/products/anymap/docs/

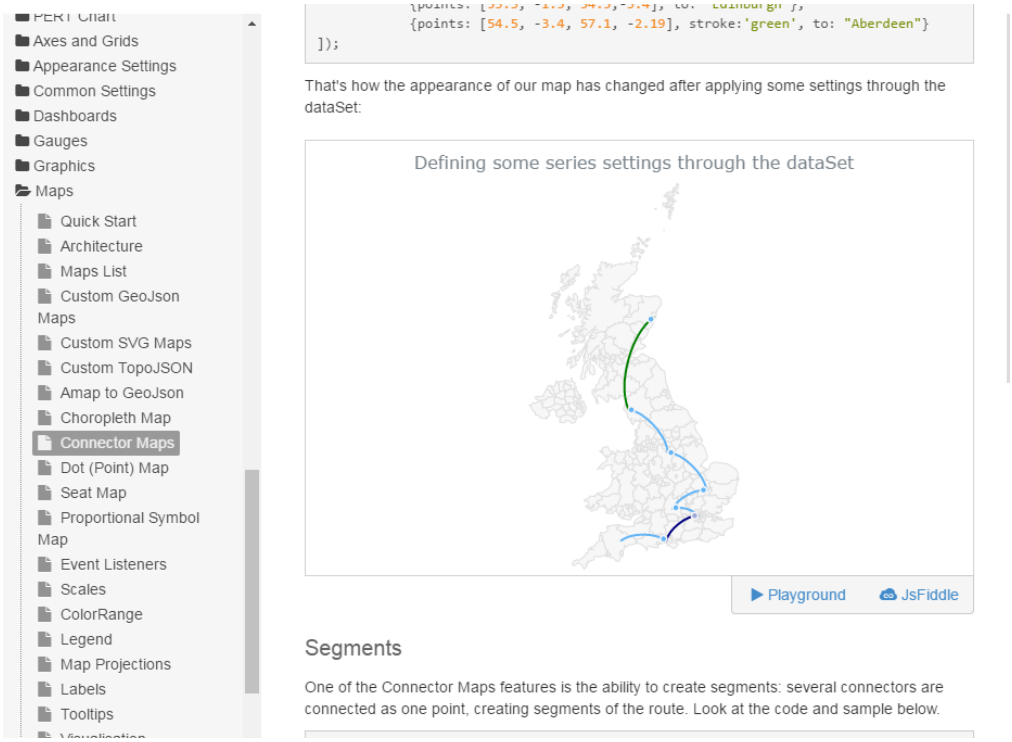


Imagen 2.26 Ejemplo de mapa geográfico de mapa de conectores generado con AnyMap. [URL]

En la parte posterior se muestra el código JavaScript que se utilizó para crear el mapa anterior.

```
anychart.onDocumentReady(function() {

  // create data set
  var data = anychart.data.set([
      {points: [50.8, -3.6, 50.8, -1.6]},
      {points: [50.8, -1.6, 51.5, -0.16], stroke:'navy', to: "New Forest National Park", marker: {fill: "#9fa8da"}},
      {points: [51.5,-0.16, 51.7, -1.29]},
      {points: [51.7, -1.29, 52.2, 0.27]},
      {points: [52.2, 0.27, 53.3, -1.5], to: "Cambridge"},
      {points: [53.3, -1.5, 54.5,-3.4], to: "Edinburgh"},
      {points: [54.5, -3.4, 57.1, -2.19], stroke:'green', to: "Aberdeen"}
  ]);
  // create map chart
  map = anychart.map()

  //create connector series
  var series = map.connector(data);

  // set geodata
  map.geoData(anychart.maps['united_kingdom']);

  //set markers
  series.markers({position: '100%', size: 3, type: 'circle'});
  series.hoverMarkers({position: '100%', size: 3, fill: '#1976d2', stroke: '2 #E1E1E1', type: 'circle'});
  series.selectMarkers({position: '100%', size: 3, fill: '#1976d2', stroke: '2 #E1E1E1', type: 'circle'});

  // name of the chart
  map.title("Defining some series settings through the dataSet");

  // set container id for the chart
  map.container('container');

  // initiate chart drawing
  map.draw();
  });
```

Imagen 2.27 Código JavaScript que se utilizó para crear el gráfico de área anterior

Anychart en general es una librería que no requiere una curva de aprendizaje tan alta como D3. Es una buena opción si se quiere aprender a utilizar código en lugar de una interfaz. Cuenta con una gran cantidad de gráficos disponibles y la posibilidad de usar mapas geográficos también, además los gráficos que se generan mediante ella son responsive. Si se requiere un soporte más personalizado la versión de pago sería la mejor opción. Un punto a favor de AnyChart también, es el Playground disponible para que podamos observar e ir probando los diferentes gráficos. Otro punto a tomar en cuenta es su amplia documentación.

2.2.3 Librería Chart.js

Chart.js es una librería JavaScript que es OpenSource[34]. Esta librería ofrece La posibilidad de crear gráficos de manera sencilla para nuestras visualizaciones de datos. Las características principales de esta librería son que los gráficos que realicemos en ella son responsive, y también se visualizarán correctamente en la mayoría de los navegadores. Además, ofrece crear animaciones dentro de los gráficos para un mayor impacto visual.

Chart.js ofrece ocho diferentes tipos de gráficos, dentro los que destacan: gráficos de barras y columnas, gráfico de línea, gráfico de radar, gráfico de araña, gráfico de pie y gráficos de burbuja.

Chart.js cuenta con un repositorio en GithHub dónde se puede buscar más información acerca de esta librería. Además, con cuenta con una amplia documentación dónde se puede encontrar cómo comenzar a utilizar Chart.js.

[34] Código abierto

Dentro de la documentación de esta librería se puede encontrar cómo realizar cada uno de los diferentes gráficos que ofrece.

Para comenzar a crear un gráfico, se necesita incluir un elemento canvas en nuestro HTML y asignarle un id.

```
<canvas id="myChart" width="400" height="400"></canvas>
```

Posteriormente se agregan las siguientes líneas de código en un archivo .js para empezar a generar el gráfico.

```
var ctx = document.getElementById("myChart");
var myChart = new Chart(ctx, {
    type: 'bar',
    data: {
```

El gráfico posterior muestra un pequeño ejemplo de un gráfico de radar generado mediante Chart.js

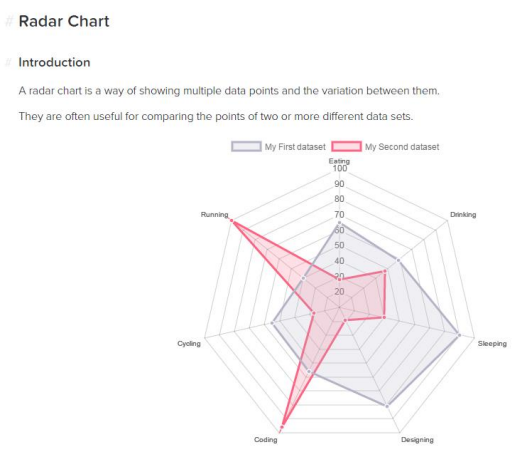


Imagen 2.28 Gráfico de radar realizado en Chart.js URL

Las líneas de código que se muestran en la parte de abajo es el código utilizado para crear el gráfico de radar antes descrito. En esta sección también se puede definir la posición del título, el tamaño del canvas donde se visualiza el gráfico, la tipografía y el tamaño de la misma que vamos a ocupa, así como el color.

```
var myRadarChart = new Chart(ctx, {
    type: 'radar',
    data: data,
    options: options
});
```

Los datos que se despliegan en nuestro gráfico se guardan en la variable "data". En nuestra variable data se pueden definir nuestras etiquetas y dentro de las opciones de los "data sets", podemos empezar a definir el color de fondo, el color de los bordes en caso de ser necesario, algunas

etiquetas complementarias y los valores que le asignaremos a los gráficos.

```
var data = {
    labels: ["Eating", "Drinking", "Sleeping", "Designing", "Coding", "Cycling", "Running"],
    datasets: [
        {
            label: "My First dataset",
            backgroundColor: "rgba(179,181,198,0.2)",
            borderColor: "rgba(179,181,198,1)",
            pointBackgroundColor: "rgba(179,181,198,1)",
            pointBorderColor: "#fff",
            pointHoverBackgroundColor: "#fff",
            pointHoverBorderColor: "rgba(179,181,198,1)",
            data: [65, 59, 90, 81, 56, 55, 40]
        },
        {
            label: "My Second dataset",
            backgroundColor: "rgba(255,99,132,0.2)",
            borderColor: "rgba(255,99,132,1)",
            pointBackgroundColor: "rgba(255,99,132,1)",
            pointBorderColor: "#fff",
            pointHoverBackgroundColor: "#fff",
            pointHoverBorderColor: "rgba(255,99,132,1)",
            data: [28, 48, 40, 19, 96, 27, 100]
        }
    ]
};
```

Imagen 2.29 Código JavaScript que se utilizó para crear el gráfico de área anterior

Chart.js solo ofrece la posibilidad de crear gráficos. Los mapas geográficos no son ofrecidos en esta librería.

La curva de aprendizaje para utilizar esta librería es de mediana a baja. Esta librería es muy buena opción en caso de que solo queramos realizar gráficos sencillos. Un punto importante de esta librería es que es de código abierto por lo que no se requiere invertir nada de dinero para poder utilizarla.

La única desventaja que pudiera tener está librería con respecto a D3 y AnyChart.js es que cuenta con una cantidad de gráficos muy limitada

comparada con las otras dos. Además de no contar con la disponibilidad de poder utilizar mapas geográficos.

Librería	D3.js	AnyChart.js	Chart.js
Costo	Código abierto	Se puede descargar gratis, aunque hay una versión de pago en caso de requerir soporte	Código abierto
Tutoriales/Repositorios	Sí	Sí	Sí
Responsive	Sí	Sí	Sí
Curva de aprendizaje	Alta	Media-baja	Media-baja
Cantidad de gráficos disponibles	Más de 50 diferentes tipos de gráficos	36 diferentes tipos de gráficos	8 diferentes tipos de gráficos
Posibilidad de usar mapas geográficos	Sí	Sí	No
Soporte	No, pero las comunidades de StackOverflow que recomiendan están muy activas	Sí, pero solo en la versión de pago. En caso de no tener licencia se puede recurrir a la comunidad de StackOverflow	No
Crossbrowser	Sí	Sí	Sí

Tabla 5. Tabla comparativa de características de librerías JS pensadas en un perfil de desarrollador Web

Las primeras tres librerías que son pensadas en un perfil de diseñador Web cuentan con una interfaz muy similar para poder crear visualizaciones de datos. Las tres tienen una opción que nos permiten utilizarlas de manera gratuita, sin embargo, en algunas de ellas como HighCharts y AM Charts esta versión gratuita incluye una marca de agua. HighCharts y Am Charts ofrecen algunos tutoriales sobre cómo empezar a utilizarlas, además de una amplia documentación dónde podemos encontrar de modo más amplio toda la funcionalidad que nos ofrecen. Por otro lado, Live Gap Charts no ofrece ningún tipo de apoyo. Sin embargo, la interfaz es fácil de utilizar y no supone un gran reto aprender a utilizarla. HighCharts y AM Charts tienen la ventaja también de que los gráficos que generemos en estás librerías son responsive.

La librería que nos ofrece la mayor cantidad de gráficos que podemos utilizar es HighCharts. Esta librería nos ofrece 11 diferentes tipos de gráficos que podemos utilizar. Por su parte AM Charts ofrece 7 diferentes tipos de gráficos y Live Gap Charts ofrece 8 diferentes tipos de gráficos.

AM Charts es la única opción de las 3 que ofrecen la posibilidad de crear mapas geográficos para la visualización de datos. En la característica de Crossbrowser, HighCharts y AM Charts son soportados por todos los navegadores actuales, y tienen buena compatibilidad con navegadores anteriores. Live Gap Charts puede llegar a tener algunos detalles con algunos navegadores.

Dentro de las tres librerías descritas, ninguna ofrece soporte en la versión gratuita, eso sería una desventaja en caso de que llegáramos a tener algún problema al utilizar esta librería.

En general las tres librerías tienen una interfaz similar, que es intuitiva y fácil de aprender a utilizarla. Además, en librerías como HighCharts o AM Charts se pueden importar datos desde hojas de cálculo como Excel, lo cual facilita aún más la realización de las visualizaciones de datos.

En general, HighCharts y AM Charts son las mejores opciones para crear visualizaciones de datos mediante una interfaz. HighCharts cuenta con una ligera ventaja al ofrecer una mayor cantidad de gráficos que podemos utilizar. AM Charts cuenta con la ventaja de poder utilizar mapas geográficos en caso de que los requiramos.

Pensando en las librerías que tienen un perfil de desarrollador Web, las librerías analizadas tienen en común que para crear las visualizaciones de datos se utiliza sólo código. Dos de ellas que son D3.js y Chart.js son completamente de código abierto. La última que es AnyChart.js ofrece una versión gratuita y una de paga. Estás librerías requieren conocimiento previo del lenguaje de programación JavaScript, sin embargo, algunas librerías son más fáciles de utilizar que otras.

Por ejemplo, las librerías AnyChart.js y Chart.js son fáciles de aprender a utilizar e implementar. La curva de aprendizaje es baja en comparación con la última librería que es D3.js. Esta librería supone invertir más tiempo para aprender a utilizarla, sin embargo D3 es la librería más completa si se aprende por completo.

Las tres librerías tienen como característica que tienen tutoriales o repositorios para empezar a utilizarlas. Además de que los gráficos que generemos con dichas librerías son responsive.

D3.js es la librería que más tipos de gráficos ofrece. Cuenta con más de 50 diferentes tipos de gráficos que están disponibles para que podamos utilizarlos. AnyChart lo sigue con 36 diferentes tipos de gráficos y en el último lugar se encuentra Chart.js con solo 8 tipos de gráficos.

Tanto D3.js como AnyChart.js ofrecen la posibilidad de utilizar mapas geográficos. Chart.js no cuenta con esta opción porque se enfoca solo en gráficos más simples.

En la característica de soporte solo AnyChart.js ofrece soporte, pero en su versión de pago. Las otras dos librerías no cuentan con soporte alguno. Sin embargo, los desarrolladores de estás librerías invitan a utilizar los apartados de StackOverflow en caso de tener algún problema con estas librerías. Las comunidades están regularmente activas y son una buena fuente de consulta. Por último, las librerías tienen buena compatibilidad con la mayoría de los navegadores.

De las tres librerías D3.js es la que ofrece un mayor número de gráficos que podemos utilizar, y es la librería más completa. Sin embargo, la curva de aprendizaje para dominarla es alta, por lo que requiere invertir mucho tiempo para utilizarla. AnyChart.js es una buena opción que cuenta con 36 gráficos disponibles y la curva de aprendizaje no es tan alta como D3.js. Por último Chart.js es una opción si solo requerimos utilizar gráficos sencillos, o como introducción a crear visualizaciones de datos sólo con código.

Conclusión

La hipótesis del capítulo era que, las librerías JS, pensando en un perfil de diseñador Web, tienen como característica principal ser más visuales en el proceso de desarrollar las gráficas, además de que tienen un costo. Por otro lado las librerías JS, pensando en un perfil de desarrollador, tienen como característica principal basarse en usar solo código, además de ser de librerías de código abierto.

Dentro de las librerías que se analizaron pensando en un perfil de diseñador Web se comprobó que tienen una interfaz similar donde se pueden crear las visualizaciones de datos de manera gráfica y al finalizar solo basta con copiar y pegar el código para utilizarlas. En la hipótesis se planteaba que estas librerías tenían un costo. Está parte resulto ser incorrecta ya que estás librerías ofrecen una versión gratuita que podemos utilizar sin cualquier inconveniente.

Por su parte las librerías que fueron analizadas pensando en un perfil de desarrollador Web sí tienen la característica mencionada: las visualizaciones, se parte solo de escribir el código. Dos de ellas son de código abierto y una ofrece una versión gratuita y una de paga dependiendo de nuestras necesidades.

Las librerías que se sugieren utilizar varían dependiendo de nuestras necesidades. Tanto si se desea utilizar una interfaz o solo escribir el código lo recomendable es probar y utilizar las diferentes librerías que se proponen en este capítulo. Una vez que se hayan probado las librerías será más fácil decidir cuál se adapta a nuestras necesidades y con cual nos sentimos más cómodos trabajando.

Por lo anterior se puede decir que la hipótesis se verifica de manera parcial porque muchas de las librerías que están pensadas en un perfil de diseñador Web tienen la opción para utilizarlas de manera gratuita, solo con el inconveniente de una marca de agua en la parte inferior. Por otro lado en las librerías que están pensadas en un perfil de desarrollador Web existe una librería (AnyChart.js) que podemos utilizar de manera gratuita o que podemos comprar en caso de necesitar soporte por parte del desarrollador de la librería.

Capítulo 3

Elaboración del sitio Web "Data Picker Guide"

El último capítulo consiste en documentar la elaboración del sitio Web "Data Picker Guide", el cual sirve como guía a diseñadores y desarrolladores que quieran crear visualizaciones de datos. El sitio Web abarca los puntos tocados en la investigación que son la elección de algún gráfico, las paletas de colores y tipografías para la visualización de datos. También se incluye un apartado de las librerías JavaScript que se analizaron en el capítulo dos.

El sitio está dividido en seis secciones principales que son: Acerca, gráficos, paletas de color, tipografía, herramientas y recursos. En la sección de acerca, se menciona el proceso de la investigación y los objetivos. En la sección de gráficos se incluyen los criterios que se concluyeron de la investigación y se mencionan también los tipos de gráficos que son recomendados por cada criterio. En la sección de la paleta de color se menciona las recomendaciones finales de la investigación además de incluir ejemplos de paletas de colores recomendadas para la visualización de datos.

Por su parte la sección de tipografía menciona las conclusiones y recomendaciones para elegir una tipografía para usar en la visualización de datos además de incluir algunos ejemplos de tipografía. En la penúltima sección, herramientas, se mencionan las librerías JS que fueron analizadas en capítulo dos, así como sus características y en qué proyectos es recomendable utilizarlas.

Por último, la sección de recursos se puede encontrar todos los elementos consultados como complemento a la investigación.

3.1 Mapa de sitio

Mapa de sitio

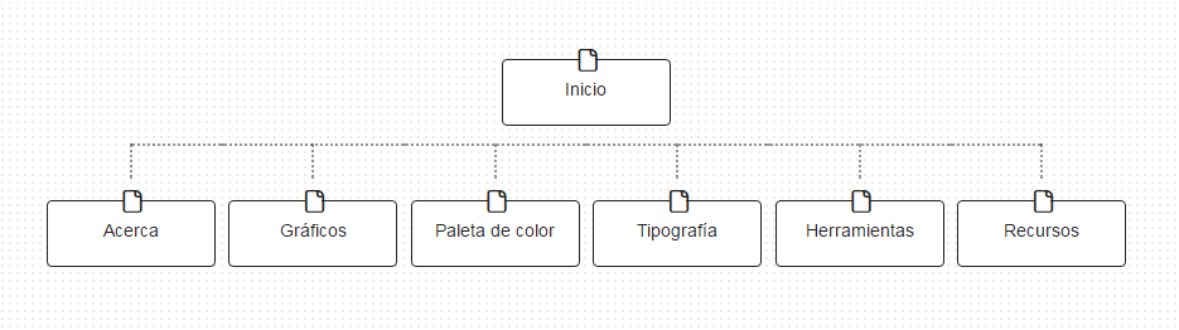


Imagen 3.1 Mapa de sitio del sitio Data Picker Guide

El sitio Web está disponible mediante las Github Pages, además de contar con un repositorio en Github para una mejor documentación.

El sitio Web está maquetada mediante el uso de las cajas flexibles (FlexBox), por lo que es totalmente responsive y se puede ver en cualquier dispositivo móvil.

3.2 Sección Acerca

Está sección es la entrada al sitio de Data Picker Guide. En ella se encuentra una breve introducción acerca de las secciones con las que cuenta la guía. También se detalla cuáles son los objetivos de la investigación.

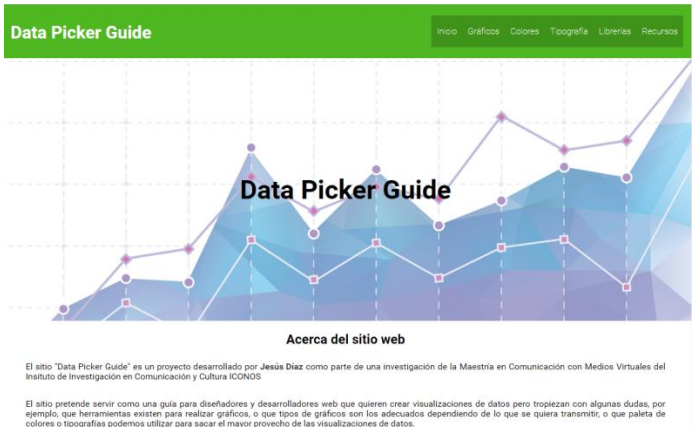


Imagen 3.2 Sitio Web Data Picker Guide URL

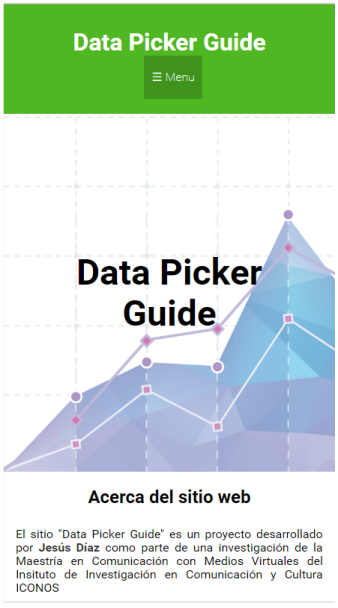


Imagen 3.3 Sitio Web Data Picker Guide en un Dispositivo Móvil

3.3 Sección Gráficos

La sección de Gráficos es la sección más extensa del sitio Data Picker Guide. En ella se describen los cuatro criterios básicos que fueron propuestos para la selección de un tipo de gráfico.

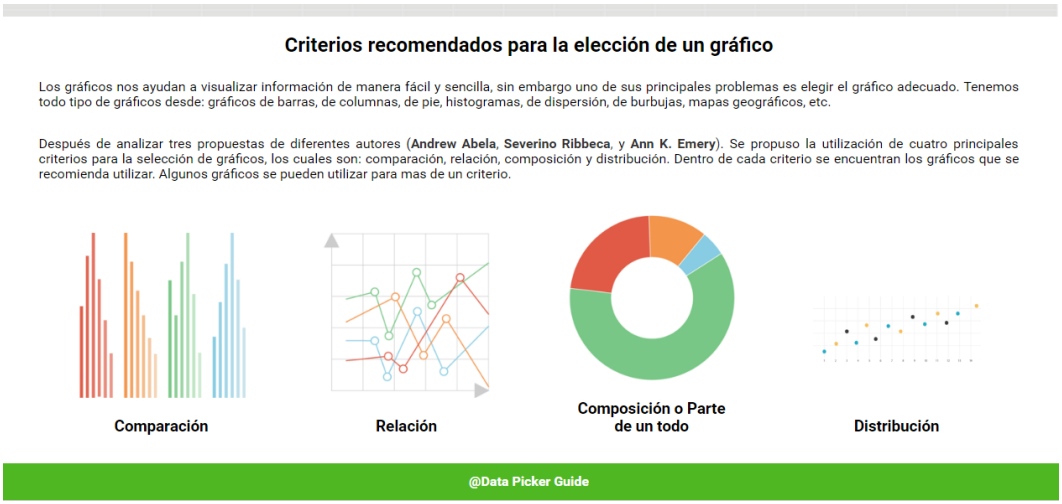


Imagen 3.4 Sección Gráficos del sitio Data Picker Guide URL

Dentro de cada uno de los criterios se encuentran los gráficos que se recomienda utilizar de acuerdo con el criterio que se haya elegido.

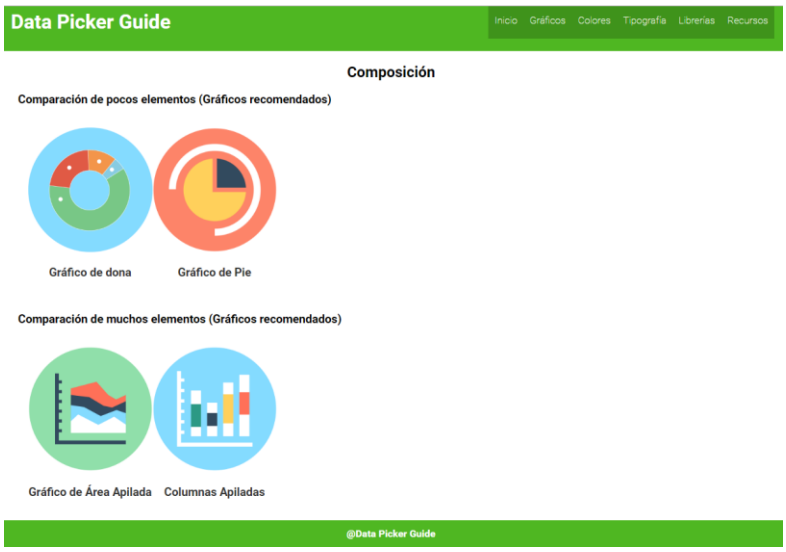


Imagen 3.5 Sección de gráficos recomendados para el criterio de Composición URL

Si entramos a alguno de los gráficos dentro de esta sección muestra un ejemplo del tipo de gráfico que hemos elegido. Además del ejemplo también se muestra una pequeña definición del gráfico y en la parte derecha se muestran algunas librerías JavaScript que podemos utilizar para la creación de ese tipo de gráfico.

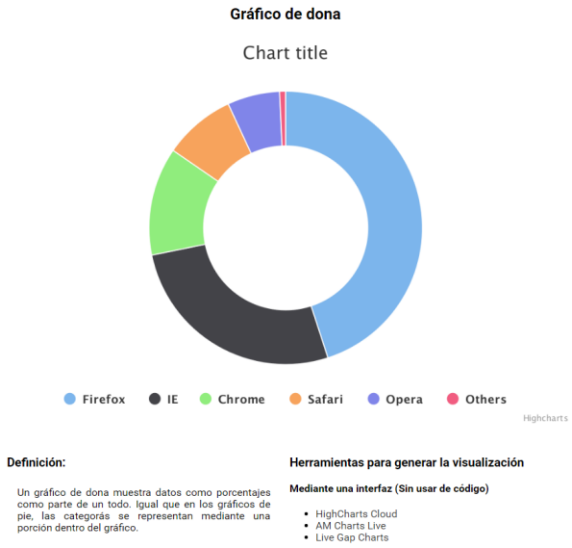


Imagen 3.6 Gráfico de dona. URL

3.4 Sección Colores

La siguiente sección que forma parte de la guía es la sección de Colores. En esta sección se encuentra el apartado de las recomendaciones que se deben tomar en cuenta al momento de elegir una paleta de colores para la visualización de datos. En la parte inferior se muestran tres paletas de colores que fueron propuestas por Samantha Zang en el Artículo "Finding the Right Color Palletes for Data Visualizations".

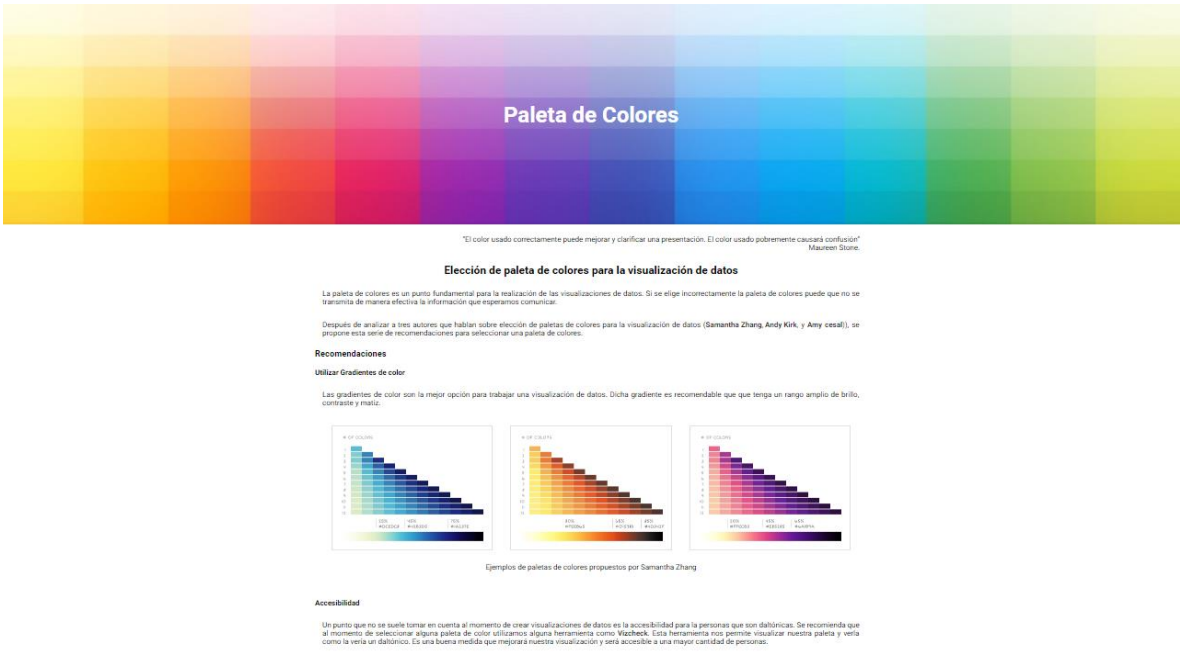


Imagen 3.7 Sección colores del sitio Web Data Picker Guide URL

3.5 Sección Tipografía

La siguiente sección es la de Tipografía. En ella se resumen las recomendaciones sobre la elección de la tipografía para la visualización de datos. En la parte posterior se muestran las seis tipografías que se recomiendan utilizar para la visualización de datos.

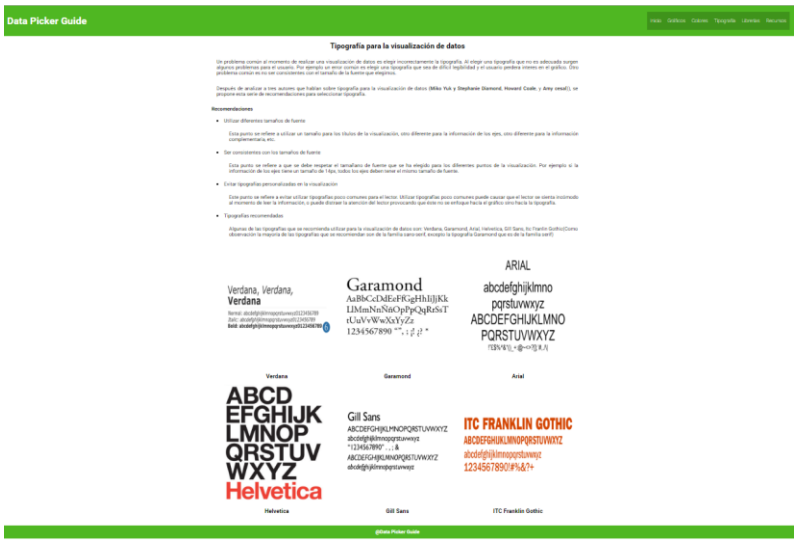


Imagen 3.8 Sección Tipografía del sitio Data Picker Guide URL

3.6 Sección Librerías

La siguiente sección es la de librerías. En esta sección se encuentra una pequeña tabla donde se resume el capítulo dos de la investigación que es el comparativo de las librerías JavaScript para realizar visualizaciones de datos.

La tabla se divide en dos secciones las cuales son las librerías que se recomiendan para los diseñadores Web y las librerías que se recomiendan para los desarrolladores Web. La tabla abarca ocho características esenciales que fueron analizadas en el capítulo dos. Estas características son: Precio, si cuentan con tutoriales o repositorios para aprender a utilizarlas, si son *responsive*, si la curva de aprendizaje es alta, la cantidad de gráficos que ofrecen, si ofrecen el uso de mapas geográficos, si dan soporte y si son *crossbrowser*.



Librerías JavaScript para la visualización de datos

En esta sección se encuentran sintetizada la investigación que se realizó de seis librerías JavaScript para la visualización de datos. Las librerías se clasifican en dos grupos, tres de ellas fueron analizadas pensando en un perfil de diseñador web y las otras tres fueron pensando en un perfil de desarrollador web.

Las librerías analizadas pensando en un perfil de diseñador tienen como característica principal que se realizan mediante una interfaz. Las librerías que fueron analizadas son: **HighCharts, AM Charts** y **Live Gap Charts**

Las librerías analizadas pensando en un perfil de desarrollador tienen como característica principal que se realizan mediante el uso de código JavaScript. Las librerías que fueron analizadas son: **D3.js, AM Charts** y **Chart.js**

Librerías JavaScript mediante una interfaz

Librería	High Charts(Cloud)	AM Charts (Live Editor)	Live Gap Charts
Costo	Gratis con marca de agua. Sin marca 15 dólares al mes	Gratis con marca de agua	Gratis
Tutoriales/Repositorios	Sí	Sí	No
Responsive	Sí	Sí	No
Curva de aprendizaje	Intuitiva y fácil de usar	Intuitiva y fácil de usar	Intuitiva y fácil de usar
Gráficos disponibles	11 diferentes tipos de gráficos, con algunas variaciones	7 diferentes tipos de gráficos con algunas variaciones	9 diferentes tipos de gráficos
Mapas geográficos	No	Sí, pero solo coroplético	No
Soporte	Sí, pero en la versión de paga	Sí, pero en la versión de paga	No
Crossbrowser	Sí	Sí	No

Librerías JavaScript con código

Librería	D3.js	AnyChart.js	Chart.js
Costo	Código abierto	Se puede descargar gratis, aunque hay una versión de pago en caso de requerir soporte	Código abierto
Tutoriales/Repositorios	Sí	Sí	Sí
Responsive	Sí	Sí	Sí
Curva de aprendizaje	Alta	Media-baja	Media-baja
Gráficos disponibles	Más de 50 diferentes tipos de gráficos	36 diferentes tipos de gráficos	8 diferentes tipos de gráficos
Mapas geográficos	Sí	Sí	No
Soporte	No, pero las comunidades de StackOverflow que recomiendan están muy activas	Sí, pero solo en la versión de pago. En caso de no tener licencia se puede recurrir a la comunidad de StackOverflow	No
Crossbrowser	Sí	Sí	Sí

@Data Picker Guide

Imagen 3.9 Sección Librerías del Sitio Data Picker Guide URL

3.7 Sección Herramientas

La última sección de la guía es la de Recursos. Dentro de esta sección se encuentran a manera de listado todas las fuentes consultadas durante la investigación de este proyecto.

El sitio Web puede ser consultado en la siguiente dirección: http://greensounds.com.mx/Data-Picker-Guide/index.html. Data Picker Guide también cuenta con un repositorio en Github donde se puede descargar el sitio en caso de que sea necesario. La dirección para el repositorio de Github es la siguiente: https://github.com/greensounds/Data-Picker-Guide

Conclusiones

Se construyó una guía digital y no impresa porque el público al cual se enfoca la guía son personas que buscan crear visualizaciones de datos para la Web. Una fuente de consulta digital se considera oportuna para el tipo de usuario que va a consultarla.

El sitio Web se divide en seis principales secciones con el objetivo de categorizar algunos de los principales problemas que tienen los desarrolladores y los diseñadores al momento de crear una visualización de datos. El objetivo del capítulo tres se logró, ya que se desarrolló de la guía, dirigida principalmente a diseñadores y desarrolladores que cuentan con nociones muy básicas del lenguaje de programación JavaScript hasta los que cuentan con un nivel más alto en éste lenguaje.

Conclusiones Finales

A lo largo del presente documento se han observado distintos aspectos interesantes.

En el primer capítulo donde se analizaron la elección de los tipos de gráficos, paleta de colores y la tipografía, fue interesante notar que los autores coincidían en los principales criterios de selección para realizar alguna visualización, también que muchos gráficos pueden servir para más de una función. En el apartado de la elección de paleta de colores fue interesante notar las características a tomar en cuenta para que las visualizaciones logren su objetivo principal Dentro de las principales características destacan elegir una paleta de colores que contenga un amplio rango de brillo y matiz. Por último en el apartado de la tipografía se hizo notar que las tipografías de la familia san serif son la mejor elección al momento de realizar una visualización, específicamente las seis familias que se presentan en ese apartado, además que hay que ser consistentes con el tamaño que seleccionemos para los diferentes elementos de la visualización.

Por lo tanto, la tesis del primer capítulo dice que:

Una vez elegido el tipo de gráfica, partiendo de los principales criterios que son: comparación, composición, distribución y relación, se puede elegir una tipografía, preferente de la familia san serif, que sea de fácil lectura para el usuario, evitando la utilización de tipografías personalizadas, y siendo consistentes con el tamaño de fuente de los elementos de la visualización. Posteriormente se podrá elegir una paleta de colores que incluya una degradación de color, con una amplia variación de brillo y matiz, y preferentemente revisando que la paleta sea accesible a las personas daltónicas.

En el capítulo dos se identificó que las librerías tienen muchas características en común, por ejemplo, las tres librerías que se tomaron en cuenta para un perfil de diseñador Web tienen una interfaz similar muy potente, además de ser muy amigables para los usuarios que apenas comienzan a realizar visualizaciones. Por su parte las librerías seleccionadas pensando en un perfil de desarrollador también compartían características similares, aunque el nivel de tiempo que se debe invertir para aprender dichas librerías es mayor.

Las seis librerías analizadas fueron solo una muestra representativa de todas las herramientas y librerías disponibles para crear visualizaciones de datos. Estas herramientas fueron elegidas por su popularidad en la Web. Existen muchas otras librerías JavaScript para realizar visualizaciones y otras herramientas que no se basan en JavaScript. Existen otras opciones interesantes que podrían ayudar a los usuarios a crear visualizaciones. Algunas de ellas son: Google Charts, Leaflet, Tableau, CartoDB, etc, el universo de herramientas es amplio.

Por lo tanto, la tesis del segundo capítulo dice que:

Las librerías JS, pensando en un perfil de diseñador Web, que fueron analizadas en esta investigación tienen como características principales una interfaz similar que se enfoca más en el aspecto visual y no tanto en la programación, además de ofrecer opciones gratuitas y de paga para sus usuarios. También ofrecen la importación de archivos externos como CSV o Excel. Por otro lado, las librerías analizadas pensando en un perfil de desarrollador Web, tienen como características principales ofrecer un mayor número de opciones de gráficos, se enfocan más en la creación de visualizaciones mediante código, y tienen opciones que son gratuitas y de paga.

Un tema que no se tocó durante esta investigación, pero tiene gran relevancia con la visualización de datos es la conexión de las librerías JavaScript con algún lenguaje del lado del servidor para generar visualizaciones de datos dinámicas. Algunos de los lenguajes de programación de lado del servidor que se pueden ocupar son: PHP, Node.js, Ruby on Rails, Python, etc. Un posible tema de investigación que se puede desprender es cómo conectar dichas librerías con el servidor y qué tan amigables son estas librerías con los lenguajes del lado del servidor. Así como qué bases de datos se pueden ocupar para esas conexiones (sql, mysql, MongoDB, etc.)

Fuentes de Consulta

Abela, Andrew V. "Chapter 7: Visuals: Visual Presentation Elements: Graphic, Charts, Color, Animations and Fonts." *Advanced Presentations by Design: Creating Communication That Drives Action*. 2nd ed. San Francisco: Pfeiffer, 2008. 111-16. Print.

Coale, Howard. "Important of Colors, Fonts and Icons." *Infoactive.co*. N.p., n.d. Web. 8 June 2016. https://infoactive.co/data-design/ch15.html

Coale, Howard. "Important of Colors, Fonts and Icons." *Infoactive.co*. N.p., n.d. Web. 8 June 2016. https://infoactive.co/data-design/ch15.htmlhttps://infoactive.co/data-design/ch15.htmlhttps://infoactive.co/data-design/ch15.htmlhttps://infoactive.co/data-design/ch15.html

Emery, Ann K. "Emery's Essentials." *Anns Blog*. N.p., 25 Feb. 2015. Web. 06 June 2016. http://annkemery.com/essentials/http://annkemery.com/essentials/http://annkemery.com/essentials/http://annkemery.com/essentials/

http://digital.csic.es/bitstream/10261/96086/1/Daltonismo_Menendez.pdf

https://github.com/amycesal/dataviz-style-guide/blob/master/Sunlight-StyleGuide-DataViz.pdf

Kirk, Andy. "Conceiving and Reasoning Visualization Design Options." *Data Visualization a Succesful Design*. N.p.: Packt, 2012. 96-106. Print.

Ribecca, Severino. "What Do You Want to Show?" *Search: The Data Visualisation Catalogue*. N.p., n.d. Web. 06 June 2016. <http://www.datavizcatalogue.com/search.html>.

Yuk, Miko, and Stephanie Diamond. "Choosing Text Fonts for Data Visualizations." *Dummies.com*. N.p., n.d. Web. 08 June 2016. http://www.dummies.com/how-to/content/choosing-text-fonts-for-data-visualizations.html

Zhang, Samantha. "Finding the Right Color Palettes for Data Visualizations." *Medium*. N.p., 23 Nov. 2015. Web. 14 June 2016. <https://blog.graphiq.com/finding-the-right-color-palettes-for-data-visualizations-fcd4e707a283#.d6lmowljz>.

Imágenes

Capítulo 1

Imagen 1.1 Diagrama realizado por Andrew Abela, tomando como base 4 criterios de selección, con el objetivo de guiar a los usuarios en la elección de una gráfica para la visualización de datos URL

Imagen 1.2 Data Visualization Catalogue, es una guía realizada por Severino Ribecca, para ayudar a seleccionar un gráfico. Consta de 16 tipos de criterios. URL

Imagen 1.3 Gráficos recomendados para el criterio de comparación basado en ejes URL

Imagen 1.4 Gráficos recomendados para el criterio de comparación no basado en ejes URL

Imagen 1.5 Gráficos recomendados para el criterio de proporción URL

Imagen 1.6 Gráficos recomendados para el criterio de proporción como parte de un todo URL

Imagen 1.7 Gráficos recomendados para el criterio de Parte de un todo. URL

Imagen 1.8 Gráficos recomendados para el criterio de relación. URL

Imagen 1.9 Gráficos recomendados para el criterio de localización. URL

Imagen 1.11 Gráficos recomendados para el criterio de distribución mostrada geográficamente. URL

Imagen 1.12 Gráficos recomendados para el criterio de movimiento. URL

Imagen 1.13 Gráficos recomendados para el criterio de patrones. URL

Imagen 1.14 Gráficos recomendados para el criterio de rangos. URL

Imagen 1.15 Gráficos recomendados para el criterio de datos que cambian con el tiempo. URL

Imagen 1.16 Gráficos recomendados para el criterio de comparación. URL

Imagen 1.17. Gráficos recomendados para el criterio de dispersión. URL

Imagen 1.18 Gráficos recomendados para el criterio de composición (parte de un todo). URL

Imagen 1.19 Gráficos recomendados para el criterio de mapas geográficos. URL

Imagen 1.20 Gráficos recomendados para el criterio de correlación. URL

Imagen 1.21 Muestra de familias Sans Serif y Serif

Imagen 1.22 Guía realizada por Amy Cesal dónde ofrece recomendaciones acerca de tamaños de fuentes y medidas de para la visualización de datos. URL

Imagen 1.23 Paletas de colores con diferente variación de brillo y contraste.

Imagen 1.24 Paleta de colores que puede ser localizada fácilmente en la naturaleza (atardecer)

Imagen 1.25 Paletas de Colores propuestas por Samantha Zhang tomando en cuenta una variación amplia de brillo, contraste y matiz.

Imagen 1.26 Ejemplo de la Andy Kirk utilizando un mapa coroplético. En el primer mapa se utilizan diferentes colores para transmitir información.

En segundo un solo color con un rango diferente de brillo, contraste y matiz.

Imagen 1.27 La paleta de colores propuesta por Amy Cesal en su guía Data Visualization Style Guidelines URL

Capítulo 2

Imagen 2.1 Lista de precios de la Licencia de HighCharts URL

Imagen 2.2 Interfaz para crear gráficos de HighCharts, llamada Cloud High Charts URL

Imagen 2.3 Proceso para crear las primeras visualizaciones de datos con el cloud de HighCharts. URL

Imagen 2.4 Segundo paso para crear visualización de datos que consiste en seleccionar los colores, tipografías, etc. URL

Imagen 2.5 Último paso para concluir la visualización el cual consiste en exportar el código JavaScript. URL

Imagen 2.6 Ejemplo de un gráfico generado mediante el Cloud de HighCharts. URL

Imagen 2.7 Lista de precios de la licencia para utilizar AM Charts. URL

Imagen 2.8 Tipos de gráficos que ofrece AM Charts. URL

Imagen 2.9 Interfaz de AM Charts Live Editor para crear visualizaciones de datos. URL

Imagen 2.10 Código JavaScript generado mediante AM Charts

Imagen 2.11 Ejemplo de gráfico de área apilada realizado con AM Charts.

Imagen 2.12 Herramienta Pixel Map de Am Charts para crear mapas geográficos. URL

Imagen 2.13 Código JavaScript generado mediante la herramienta Pixel Map

Imagen 2.14 Código Ejemplo de mapa geográfico generado con Pixel Map

Imagen 2.15 Interfaz de la herramienta Live Gap Charts para crear visualizaciones de datos. URL

Imagen 2.16 Ejemplo de gráfico generado con Live Gap Charts

magen 2.17 Apartado de tutoriales disponibles en D3.js URL

Imagen 2.18 Galería de ejemplos que tiene D3.js para consultar diferentes tipos de gráficos. URL

Imagen 2.19 Ejemplo de gráfico expuesto en la galería de D3 URL

Imagen 2.20 Ejemplo de gráfico de burbuja generado con la librería D3.js

Imagen 2.21 Ejemplo de mapa geográfico generado con la librería D3.js URL

Imagen 2.22 Lista de precios de licencia para utilizar la librería AnyChart.js URL

Imagen 2.23 Playground de la librería AnyChart.js URL

Imagen 2.24 Ejemplo de gráfico de área apilada realizado con AnyChart.js URL

Imagen 2.25 Código JavaScript que se utilizó para crear el gráfico de área anterior

Imagen 2.26 Ejemplo de mapa geográfico de mapa de conectores generado con AnyMap. URL

Imagen 2.27 Código JavaScript que se utilizó para crear el gráfico de área anterior

Imagen 2.28 Gráfico de radar realizado en Chart.js URL

Imagen 2.29 Código JavaScript que se utilizó para crear el gráfico de área anterior

Capítulo 3

Imagen 3.1 Mapa de sitio del sitio Data Picker Guide

Imagen 3.2 Sitio Web Data Picker Guide URL

Imagen 3.3 Sitio Web Data Picker Guide en un Dispositivo Movil

Imagen 3.4 Sección Gráficos del sitio Data Picker Guide URL

Imagen 3.5 Sección de gráficos recomendados para el criterio de Composición URL

Imagen 3.6 Gráfico de dona. URL

Imagen 3.7 Sección colores del sitio Web Data Picker Guide URL

Imagen 3.8 Sección Tipografía del sitio Data Picker Guide URL

Imagen 3.9 Sección Librerías del Sitio Data Picker Guide URL

Tablas

Tabla 1. Comparativa de los criterios propuestos por los tres autores

Tabla 2. Tabla comparativa de las recomendaciones que nos proponen los 3 autores

Tabla 3. Tabla comparativa de las recomendaciones que nos proponen los 3 autores con respecto a la elección de la paleta de color

Tabla 4. Tabla comparativa de las librerías pensando un perfil de diseñador Web.

Tabla 5. Tabla comparativa de características de librerías JS pensadas en un perfil de desarrollador Web

Esquemas

Esquema 1. Criterios y gráficos que se propone utilizar para la visualización de datos

Esquema 2. Esquema que contiene las síntesis de las recomendaciones propuestas por los tres autores

Esquema 3. Esquema contiene la síntesis para la elección de la paleta de colores para la visualización de datos.