



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
IIC2026 — Visualización de Información

Tarea 2

D3.js básico

Fecha de inicio: **Lunes 27 de marzo** a las **20:00:00** hrs

Fecha de entrega: **Lunes 3 de abril** a las **20:00:00** hrs

Evaluación en el contexto del curso

Esta evaluación es de naturaleza **sumativa** e **individual**, y pretende rescatar evidencias del desarrollo de ciertos resultados de aprendizaje. Por eso, tras su realización y entrega, recibirás retroalimentación sobre su desempeño y una nota que lo refleja. El material del curso relevante para esta evaluación es el siguiente:

1. **Tecnologías web:** Introducción práctica a las tecnologías web HTML, CSS y JavaScript.
2. **SVG:** Formato de gráfico vectorial y bidimensionales para diseñar las visualizaciones.
3. **Librería D3.js:** Librería de bajo nivel especializada para la creación de visualizaciones de información en conjunto a tecnologías web.

1. D3.js básico

En esta evaluación se busca que apliques los contenidos relacionados al desarrollo web y la uso de D3.js para la construcción de una pequeña visualización en SVG. Para esto, deberás replicar una imagen utilizando **únicamente** y **exclusivamente** Javascript y D3.js para cargar los archivos con la información y confeccionar el SVG correspondiente.

Se provee un archivo HTML, uno de CSS y uno de Javascript base que deberás completar con la construcción de la imagen. **No se debe modificar nada de los archivos entregados a excepción de:**

1. La sección que dice “Completar con su nombre” dentro del archivo HTML para incluir tu nombre y apellido. De este modo identificarte como el/la autor/a de esta evaluación.
2. El archivo JS. Todo a excepción de las primeras 2 líneas.
3. La sección que dice “COMPLETAR CON EL CSS ADICIONAL QUE CONSIDERES NECESARIO” dentro del archivo CSS para entregarle estilo a los elementos de la visualización. **Queda a tu criterio si deseas editar este archivo.**

En esta evaluación se utilizará un dataset construido por [Stack Overflow](#) el 2022. Cada año se entrevistan a más de 50.000 personas y se les realiza diferentes preguntas relacionadas a la computación: como el lenguaje de programación que ocupan, experiencia, herramientas que les gustaría ocupar, etc.

Un ejemplo del gráfico a construir es el siguiente:

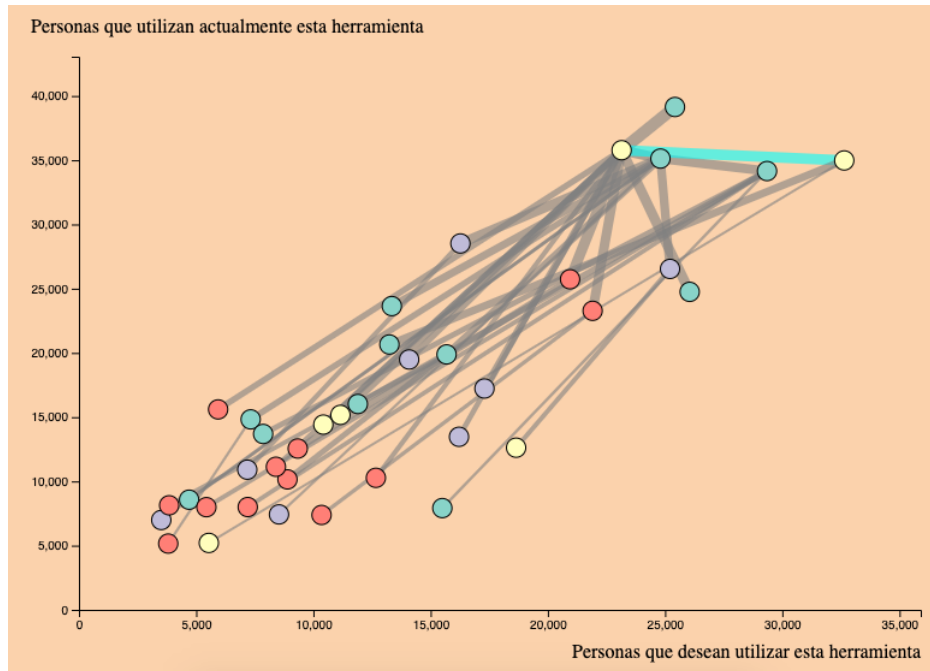


Figura 1: Ejemplo de la visualización a construir

Este gráfico permite observar diferentes lenguajes de programación, *frameworks*, tecnologías y bases de datos utilizados en la computación. En relación a los aspectos visuales del gráfico:

- El eje X codifica el atributo *desean*, que representa la cantidad de personas que desean ocupar estas herramientas.
- El eje Y codifica el atributo *trabajando*, que representa la cantidad de personas que actualmente trabajan con dicha herramienta.
- El color codifica el atributo *categoría*, que nos permite diferenciar entre lenguajes, *frameworks*, tecnologías y bases de datos. **Queda a tu criterio los colores utilizados mientras sean categóricos¹.**
- Adicionalmente, se incluye un borde negro a cada punto para poder diferenciarlos cuando hay algunos superpuestos.

Adicionalmente, se dispone de la cantidad de personas que trabajan con 2 herramientas simultáneamente. Esta información es codificada mediante líneas que unen 2 puntos en el gráfico, las cuales quedan **detrás** de los puntos. El ancho de la línea se utilizar para codificar la cantidad de personas que usan ambas herramientas simultáneamente. Mientras más ancha es la línea, más personas utilizan ambas herramientas. **El color y opacidad de las líneas queda a su criterio.** Finalmente, esta línea cambia de color si es que el cursor (*mouse*) está encima de esta.²

¹No usar diferentes gamas de un solo color. Por ejemplo: celeste, azul, azul marino y azul pastel.

²Se recomienda investigar *hover* en css.

Para construir esta imagen, se proveen de 2 *datasets* que contienen los datos necesarios. Los *links* a los archivos está incluido dentro del archivo JS entregado.

- La constante `HERRAMIENTAS_DATABASE` apunta el *dataset* que incluye la información de cada punto en el gráfico.
- La constante `TUPLAS_DATABASE` apunta al *dataset* que incluye la información de los enlaces del gráfico. Los números asociados a `nodo_1` y `nodo_2` hacen referencia al ID de cada punto en el primer *dataset*.

Al momento de entregar esta evaluación, debes asegurarte de utilizar estos *links* para cargar los datos.

Importante: en esta tarea se evaluará el correcto uso de funciones vistas en clases como `d3.json`, `d3.csv`, escalas en D3, `.attr`, `.append`, `.data`, etc. Además, se evaluará el correcto uso de los elementos SVG según corresponde, el uso de color, el correcto posicionamiento de los elementos y el correcto uso de sus atributos. **No se espera que la visualización confeccionada sea exactamente igual a la imagen mostrada, solo es una base de una posible solución.**

En cuanto a consideraciones mínimas a considerar en tu implementación, están las siguientes:

- Tu programa solo puede hacer uso de funciones nativas de JavaScript o provistas por D3.js. No se permite utilizar otras librerías de JavaScript.
- Debes utilizar los datos entregados para construir la visualización.
- Puedes hacer uso tanto de la versión 6 o 7 de D3.js.
- Se espera que utilices estilamiento nativo mediante CSS escrito por ti para esta evaluación, es decir, no se permite importar *frameworks* o usar herramientas de estilamiento ya construidas.
- Las visualizaciones deben ser desarrolladas con elementos propios de un SVG (`rect`, `circle`, `line`, `text`, `ellipse`, `g`, `path`).
- No utilizar *loops* (`for`, `while`, `forEach`, `each`) para construir la visualización. Solo si necesitan calcular algún dato adicional, se permitirá utilizar *loops*. Se recomienda confirmar su uso en el foro antes de entregar.
- Utilizar *data joins* de D3 para asociar elementos visuales a información del *datasets*.

El no seguir alguna de estas consideraciones mínimas producirá que tu evaluación no sea corregida y se califique con nota mínima.

2. Corrección y rúbrica de evaluación

Para la corrección de esta evaluación, se revisará la visualización confeccionada y el código de Javascript utilizado. Además, se usará una pauta como guía. Cada ítem será calificado de forma ternaria: cumple totalmente el ítem, cumple parcialmente el ítem, o no cumple el ítem. Además, esta pauta incluye dos ítems de descuento que serán calificados de forma binaria: hay o no hay descuento asociado según corresponda. Pueden encontrar el detalle de cada ítem a evaluar [en el siguiente spreadsheet](#), en la hoja llamada "Tarea 2".

Además de determinar el nivel de logro alcanzado, el equipo docente adjuntará retroalimentación escrita que complemente la corrección cuando sea necesario.

3. Entregables

Se espera que el entregable corresponda a: **la imagen reconstruida con elementos del SVG a partir de código JS y D3.js**. Este resultado debe entregarse como un ZIP con tres archivos: el **HTML**, el **JS** y el **CSS**. No se aceptarán entregas en cualquier otro formato distinto al indicado anteriormente (archivos TXT, DOC, etc.). De no entregar o entregar un formato diferente al especificado, no se revisará la entrega y se colocará nota mínima. **Tampoco debe incluir los datasets en el ZIP.**

4. Dudas

Cualquier duda que tengas sobre esta evaluación, prefiere publicarla en el [Syllabus del curso](#) correspondiente a esta evaluación. También, siente la libertad de responder dudas de tus pares si crees que conoces la respuesta. En caso de tener dudas que impliquen mostrar tu solución o partes de ella, no utilice este medio de consulta. Para estos casos, envíe un correo al cuerpo docente o muestre su solución solo en reunión personal (remota o presencial) cuando se reúna con algún miembro del cuerpo docente.

5. Política de atraso

Existe la posibilidad de entregar esta evaluación con hasta **3 días de atraso** a partir de la fecha de entrega definida en el enunciado. En la eventualidad de entregar pasada la fecha de entrega, se aplicará una **reducción** a la nota máxima que podrás obtener en esta evaluación.

De haber atraso, **la nota máxima a obtener** se reduce en **0.5 puntos (5 décimas)** por cada día de atraso. Cada día de atraso se determina como el techo de días de atraso. Por lo tanto, en caso de entregas atrasas, la nota final de la tarea se calcula mediante la siguiente fórmula:

$$\text{mín}(7 - (0.5 \times \text{días_atraso}), \text{nota_obtenida}) + \text{bonus}$$

Por otro lado, entregas con más de 3 días (72 hrs) de atraso no serán recibidas y serán evaluadas con la **calificación mínima (1.0)**.

6. Flexibilidad de entrega

En la eventualidad de que tengas problemas personales durante el plazo de esta evaluación, a tal punto que impida su realización de forma importante y **requieras más de los 3 días de atraso permitidos**, siéntete libre de contactar a alguien del equipo docente para buscar apoyo y opciones de flexibilidad. Para casos médicos, recuerda que antes de todo debes justificar con tu unidad académica para respetar el conducto regular de la Universidad.

Es completamente posible otorgar una extensión, de plazo individual o re-evaluar la política de atraso para cada caso en particular. Se espera que escribas explicando tu situación, al punto que sientas comodidad de hacerlo, para así entender y considerar tu caso. También se aprecia si se propone una cantidad de extensión a necesitar dentro de la solicitud.

Para estos flexibilidad posterior a los 3 días de atraso, escribir al docente del curso (hvaldivieso@ing.puc.cl).

7. Bonus de la evaluación

Esta evaluación presenta un *bonus* para bonificar la nota final de esta evaluación. Para poder optar al beneficio que entrega cada *bonus*, este debe estar implementado **en su totalidad**, es decir, **no se dará puntaje intermedio**. Adicionalmente, **todos los *bonus*** requieren que la nota en tu tarea (sin ningún tipo de *bonus*) debe ser **igual o superior a 4.0**.

Finalmente, las bonificaciones **se aplican después de la nota definida por el descuento por atraso**.

7.1. Leyenda (2 décimas)

Mediante el uso de D3.js debe incorporar, dentro del SVG, una leyenda para identificar la categoría asociada a cada color de los elementos.

7.2. Tooltip (1 décima)

Mediante el uso de D3.js debe incorporar un *tooltip* que muestre el nombre de la herramienta mientras el cursor (*mouse*) esté encima de un punto. Para esto, se recomienda investigar sobre el tag `<title>` en SVG.

7.3. Grilla (2 décimas)

Mediante el uso de D3 debe incorporar, en el fondo de la visualización, una grilla compuestas por líneas punteadas y con una baja opacidad. A continuación se muestra un ejemplo de imagen que incluye dicha grilla.

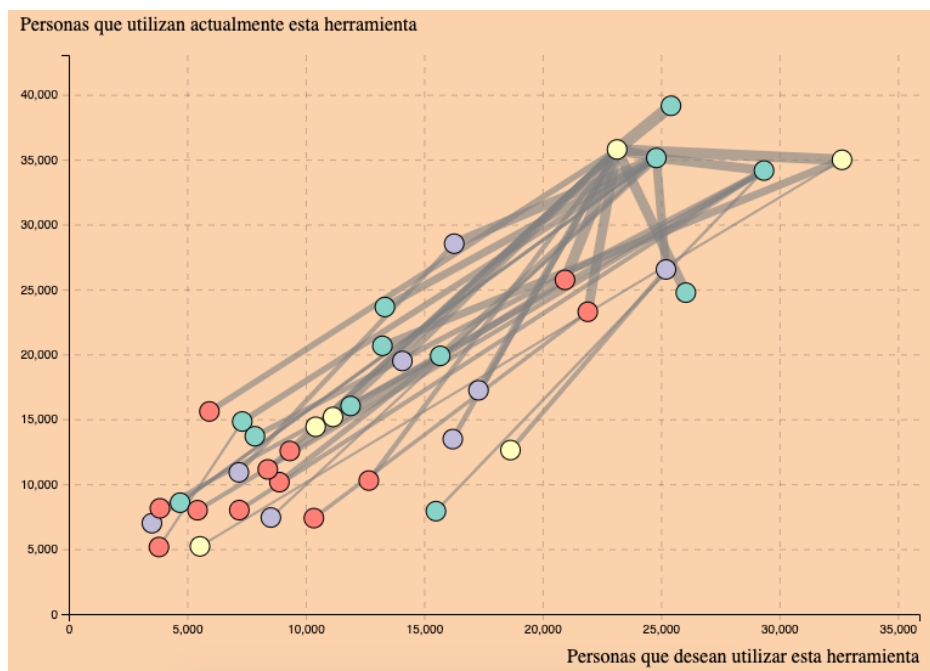


Figura 2: Ejemplo con grilla