



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

## Proyecto - Entrega 2 de 2 (60%)

### Desarrollo de una herramienta de visualización

Fecha de inicio: **22 de mayo** a las **20:00** hrs

Fecha de entrega: **24 de junio** a las **20:00** hrs.

Fecha máxima de entrega (con atraso): **25 de junio** a las **20:00** hrs.

Distribución de puntaje: [En este enlace.](#)

**Esta evaluación solo presenta 1 día como política de atraso.**

### Evaluación en el contexto del curso

Esta evaluación es de naturaleza **sumativa**, y pretende rescatar evidencias del desarrollo de ciertos resultados de aprendizaje. Por eso, tras su realización y entrega, recibirá retroalimentación sobre su desempeño y una nota que lo refleja.

Específicamente, esta evaluación se relaciona con el resultado de aprendizaje indicado en el programa del curso: **Resolver necesidades de comunicación de información mediante la programación de herramientas de visualización interactivas y Diseñar una herramienta de visualización de información mediante un modelo de trabajo, en una situación específica y escogida por el alumnado..**

Esta entrega consiste en la segunda parte del proyecto: confección de la herramienta de visualización.

## Confección de una herramienta de visualización

La intención de esta segunda parte es la confección de una página web que contenga la herramienta diseñada en la primera entrega del proyecto. No obstante, esta entrega no estará condicionada a la propuesta del proyecto, es decir, si consideran necesario cambiar una o más decisiones de diseño, se permitirá siempre y cuando cumpla con las condiciones mínimas a evaluar en esta entrega. No es necesario actualizar estos cambios en su informe.

Cómo se explicó en la primera entrega, el resultado de todo el proyecto es desarrollar **dos entregas**: un **informe en formato HTML** donde se explique el proceso realizado y las decisiones tomadas en el camino; y **una herramienta de visualización de información** implementada por ustedes, que sea el reflejo de su proceso de diseño. Para esta segunda parte, se espera únicamente la entrega del segundo resultado: la herramienta de visualización.

Nuevamente, el tema a abordar y el *dataset* que usarán como base para su experiencia puede ser escogido por ustedes. Pueden mantener el informado en la entrega anterior o bien optar por un nuevo *dataset*.

El requisito primordial de esta evaluación es la originalidad. **No se permite replicar una herramienta confeccionada por terceros**. Pueden utilizar los mismos datos que otra herramienta, pero las visualizaciones a programar deben ser diferentes. Un aspecto vital de esta evaluación es que **la herramienta a confeccionar debe tener su propia contribución substancial a la visualización de información**.

El resto de este enunciado explicita los pasos a seguir en esta experiencia, requisitos a cumplir y lo que se espera que entregues como resultado final.

### 1. Elección del tema

El primer paso es escoger la temática a tratar y el (o los) *dataset*(s) a utilizar.

Este puede ser el informado en la entrega 1 o bien modificarlo para esta entrega final. En caso de realizar un cambio, no es necesario entregar un nuevo informe, solo confeccionar un documento HTML que contenga las visualizaciones y todos los requisitos mínimos que se informan en este enunciado.

En caso de cambiar el tema y/o datos, recuerden que la única restricción a considerar sobre la elección de tema y datos, es que no es posible elegir un *dataset* o tema ya trabajado **con D3** en otra instancia del curso (Demos de clase, Ayudantías o Tareas). Al momento de elegir el *dataset*, se recomienda registrar este en el formulario indicado a continuación. De este modo, si en una ayudantía **posterior** a haber respondido el formulario se ocupa el mismo *dataset* registrado, se permitirá su uso de todas formas. Pueden responder las veces que quieran el formulario para actualizar el *dataset* a ocupar.

Link al formulario: <https://forms.gle/DiSuTByQngMCjH2x6>

Fuera de lo anterior, **no hay límites en los temas a abordar**, mientras lo traten y presenten con respeto. Como hemos visto a lo largo del semestre, ¡las visualizaciones de información permiten contar historias de cualquier tipo!

Si tienen alguna duda sobre qué tema puedes trabajar, pregunten en las [discussions del Syllabus](#).

## 2. Entregable: Herramienta

El resultado final a entregar de esta entrega es una herramienta de visualización. Esta debe estar contenida en un único documento HTML, que se extiende mediante hojas de estilo CSS, y se implemente mediante un programa escrito en **JavaScript** que se base **únicamente** en [D3.js](#).

En caso de agregar estilo a dicho documento con el fin de hacerlo visualmente más agradable y legible, Debes aplicar dicho estilo mediante un archivo externo de extensión **CSS**.

Finalmente, recordemos que en la primera entrega se trabajaron los primeros 3 niveles del *framework*, por lo que ahora solo cubriremos el último nivel.

### 2.1. Cuarto nivel: Implementación algorítmica

En base a las decisiones de codificación y diseño realizadas, se debe trabajar el cuarto nivel de: **Implementación algorítmica**. Es decir, **hacer realidad la herramienta de visualización mediante programación**. No obstante, si desean cambiar las decisiones de diseño y/o el tema informado en la entrega anterior, también se permitirá mientras la herramienta entregada cumpla con ciertos requisitos mínimos informados en este enunciado.

#### Requisitos de auto-contención

En primer lugar, como una buena interfaz humano-computador, se espera que **no presente errores de funcionamiento o en consola**, además de ser una **herramienta auto-contenida** con los elementos contextuales necesarios. Respecto a este último punto, se evaluarán 4 requisitos:

- **Título y descripción del tema:** El documento debe presentar un título acorde al tema de la herramienta, contexto de los datos y descripción general de la herramienta, de modo que toda persona pueda entender en qué consiste la herramienta.
- **Auto-contención básica de visualización:** Cada visualización debe incluir elementos mínimos para entenderla correctamente: título descriptivo, leyenda de los canales (tamaño, color, forma, etc) y nombre descriptivo de ejes (en caso de tener ejes).
- **Mención de interacciones:** Cada visualización debe explicar brevemente todas las posibles interacciones que puede realizar el usuario. Esto solo aplica si la visualización presenta interacciones. Por ejemplo, si hay *zoom* o *brush*, selección de algún elemento, si alguna manipulación afecta a otra visualización, si existen botones o selectores que afectan a la visualización, alguna acción con el *mouse*, etc. Se recomienda ver [este ejemplo](#) o [este otro ejemplo](#) donde la visualización presenta una explicación de las interacciones posibles antes de mostrar el gráfico.
- **Referencias:** Al final de la herramienta se debe incluir las **referencias a todo código utilizado que no sea creación total del estudiante**. Con incluir el enlace o mencionar las clases y/o ayudantías en las cuales se utilizó el código es suficiente. En caso de no usar ningún código de internet, en esta sección se debe declarar explícitamente que todo material es creación del estudiante/grupo.

**Importante:** Si se detecta que el estudiantado no citó algún código externo utilizado en la tarea, se penalizará en el ítem de puntaje y además se aplicará un descuento adicional por no incluir referencia al código que no es de su autoría.

## Requisitos mínimos de la herramienta

En segundo lugar, se evaluará que la herramienta contenga implementado correctamente los siguientes requisitos mínimos:

1. Incluir navegación (uso de `d3.zoom` o `zoom` semántico) y/o la posibilidad de seleccionar múltiples datos mediante el uso de `d3.brush` en al menos una visualización.
2. Incluir un filtro para **reducir** la cantidad de información desplegada en al menos una visualización. Este filtro **no puede ser** gatillado por el *zoom*. Este requisito se debe implementar mediante el uso `<input/>` de HTML o interactuando directamente con la visualización. Con este punto se espera poner en práctica un correcto uso del conjunto **exit** de D3 cuando se eliminan elementos de la visualización.
3. Incluir una interacción con el *mouse* para **seleccionar un elemento (ítem, atributo, categoría, etc.) en al menos una visualización**. El efecto de dicho evento queda a tu criterio. Esta interacción debe ser distinta a la gatillada por la multiselección de datos (*brush*).
4. Incluir transiciones para crear, actualizar y eliminar cualquier elemento de la visualización.
5. Incluir vistas coordinadas entre todos los gráficos confeccionados, es decir, deben existir suficientes interacciones que comuniquen 2 o más visualizaciones hasta cubrir el total de visualizaciones confeccionadas. Como referencia, en la tarea 2 se implementó vistas coordinadas porque se pedía una interacción donde la visualización de personajes se actualizara según la serie seleccionada en la primera visualización.
6. Programar completamente, al menos, **3 visualizaciones diferentes**. En caso de realizar el proyecto de forma **individual**, este requisito cambia a **2 visualizaciones**.

Respecto a las visualizaciones a programar en este punto, se debe cumplir las siguientes condiciones:

- (a) Ninguna de estas visualizaciones puede ser una tabla de datos o una visualización artística. Además, no se considerará como visualización un *tooltip* o un selector sofisticado. Por ejemplo, desplegar un cuadro de texto será tratado como una tabla de datos. Del mismo modo, un mapa del mundo que permita seleccionar países pero no aporte información adicional será considerado como un selector, no una visualización de información.
- (b) De las  $N$  visualizaciones a programar ( $N=3$  para proyectos en parejas y  $N=2$  para proyectos individuales), las **primeras  $N-1$  deben pertenecer al siguiente conjunto de visualizaciones**:

### **Visualizaciones para datos geográficos (máximo 1 gráfico de este conjunto en el proyecto)**

Mapa de Coropleta, Mapa de Puntos/Burbujas, Cartograma, Mapa de Flujo, Mapa de Grilla, Mapa con *Hexbins*. Mapa de calor.

### **Visualizaciones para datos de redes y/o jerárquicos (máximo 1 gráfico de este conjunto en el proyecto)**

Matriz de Adyacencia, Diagrama Nodo Enlace, Diagrama de Arco, Diagrama de Cuerdas, Diagrama de *Sankey*, *Sunburst*, *Circular Packing*.

### **Visualizaciones para datos tabulares (sin restricción a la cantidad de gráficos)**

Mapa de Calor, Gráfico que utilice simulación de fuerza, *Treemap*, Gráfico de Barra Agrupado,

Gráfico de Multilínea, Gráfico de Caja, Gráfico de Densidad, Gráfico de Violín, *Swarm Plot*, Gráfico con Ejes Paralelos, Gráfico con Ejes Circular.

- (c) La visualización número N, que es la última que debes programar, no tiene restricciones particulares, **siempre y cuando sea significativamente diferente de las demás visualizaciones**, y sea una visualización de información. No se aceptará un selector sofisticado, una tabla, una visualización artística o un *tooltip* como la visualización número N. En esta etapa, tienes la libertad de reutilizar alguna de las visualizaciones previas presentadas en clases y/o ayudantías. Sin embargo, es importante destacar que esta última visualización debe ser elegida después de haber programada las N-1 visualizaciones pertenecientes al conjunto indicado anteriormente.
- (d) **El incumplimiento de esta restricción en las visualizaciones resultará en una penalización de 3 puntos, equivalente a 30 décimas, que se descontarán de la nota final del proyecto.**

Es fundamental garantizar que las primeras N-1 visualizaciones correspondan a alguna de las presentadas en el punto 6.b. Se recomienda realizar una *discussion* o confirmar con el cuerpo docente en caso de no estar completamente seguros de este aspecto. En caso de incumplimiento de esta restricción, no se aceptarán argumentos basados en el desconocimiento, tales como: “yo pensé que mi visualización si estaba en el catálogo”

Finalmente, si por cualquier motivo se encuentran en la situación de tener que implementar menos de las N visualizaciones solicitadas, es imperante que las visualizaciones creadas pertenezcan al catálogo expuesto en el punto 6.b del enunciado. Recuerden que no cumplir con esta instrucción resultará en una penalización de 3 puntos, equivalente a 30 décimas, en la nota final del proyecto. Para esta parte del proyecto, es imperativo que comiencen desarrollando las visualizaciones expuestas en el punto 6.b antes de proceder con la visualización libre.

## Requisitos de código

En tercer lugar, en esta etapa de implementación se evaluará un correcto uso de D3.js. Por lo tanto, el código entregado debe cumplir con los siguientes requisitos:

1. **Utilizar exclusivamente D3:** se debe utilizar únicamente D3.js para la carga de datos, selección de elementos del SVG, actualizar elementos del SVG, obtener información de algún elemento, realizar transiciones, vincular eventos y eliminar elementos del SVG. En resumen, utilizar D3 en todo momento en vez de recurrir directamente al DOM (evitar el uso de `document`).
2. **Data join:** Utilizar en todo momento y **correctamente** *data join* para crear, actualizar y eliminar elementos de la visualización. Por ejemplo, hacer `d3.selectAll(...).remove()` para limpiar una visualización es un uso **incorrecto**.
3. **Escalas de D3:** Utilizar correctamente escalas de D3 (numéricas, categóricas y/o color) para transformar cualquier información del *dataset* (atributo y/o posición del dato) en algún atributo de los elementos del SVG.
4. **Código robusto:** Todo el código debe ser construido de modo que si se alteran los valores de algún atributo y/o se modifica la cantidad de elementos del *dataset*, el código y la visualización sigan funcionando correctamente. Solo se revisarán que el código y las visualizaciones funcionen mediante cambios que no contradigan el formato expuesto del *dataset*.
5. **Ausencia de errores:** El código no presenta errores en consola y de funcionamiento.

## Requisitos técnicos (descuentos en caso de no cumplir)

Finalmente, al momento de implementar la herramienta, se debe tener en cuenta las siguientes consideraciones técnicas:

- Cada evento en la herramienta debe tomar como máximo **5 segundos**. Se entiende por evento: cargar la visualización, filtrar el gráfico, seleccionar un dato, entre otros.
- Toda la herramienta debe estar contenida en un **único** HTML que presente todas las visualizaciones y textos necesarios de auto-contención.
- Se debe utilizar las versiones 6 o 7 de D3.js para el desarrollo de la herramienta. No se revisarán visualizaciones que ocupen otra versión o no ocupen D3 para su creación.
- Las visualizaciones deben ser desarrolladas únicamente con elementos propios de un SVG. En particular, se deben usar únicamente elementos vistos en clases (`rect`, `circle`, `text`, `ellipse`, `path`, `line`, `g`).
- No utilizar *loops* (`for`, `while`, `forEach`, `each`) para construir, actualizar y/o eliminar elementos de la visualización. Únicamente si necesitan calcular algún dato o filtrar el *dataset*, se permitirá utilizar *loops*. Se recomienda confirmar su uso en el foro antes de entregar. No se revisarán visualizaciones que ocupen *loops* para la creación de sus elementos y/o elementos de auto-contención.
- Se espera estilamiento nativo mediante CSS para esta evaluación, es decir, no se permite importar *frameworks* o herramientas de estilamiento ya construidas. Además, el documento HTML no debe contener *css-inline* o *internal-css*.
- El programa solo puede hacer uso de la librería D3.js. Pueden existir excepciones a esta regla, como la librería de manejo de TopoJSON o la librería de D3 para hacer Hexbin. De tener la duda sobre cierta librería, pregunta en las [discussions del Syllabus](#).

**De no seguir alguna de estas especificaciones, esta entrega presentará un fuerte descuento.**

## 3. Corrección y pauta de evaluación

Para la corrección de esta evaluación, se revisará el resultado de la herramienta y del código entregado. 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; o de forma binaria: cumple totalmente o no.

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

Finalmente, para poder ejecutar cada proyecto que tenga datos locales, se utilizará la extensión [live-server de Visual Studio Code](#) para levantar un servidor y así poder interactuar con la herramienta. Por este motivo, se recomienda utilizar [live-server de Visual Studio Code](#) para ir probando esta entrega.

## 4. Entregables

Se espera como entregable un documento **HTML** extendido mediante archivos **CSS**, JS necesarios, junto con los datos a utilizar.

La entrega se efectuará a través de un archivo comprimido en formato ZIP que debe contener tanto todo lo necesario para una correcta interacción de la herramienta **junto a los datos que utilizarán**. Esta entrega

se realizará a través de la plataforma Canvas, en la sección correspondiente a la evaluación. Tienes la flexibilidad de hacer tantas entregas como consideres necesarias, y el cuerpo docente revisará la última de ellas. **En caso de ser parejas, solo uno debe entregar.**

Cualquier otro archivo (imágenes, hojas de estilo, *scripts*, **datos**, etc.) que sea necesario para la visualización correcta de tu entrega **debe también ser incluido en el archivo comprimido.**

## 5. Dudas

Cualquier duda que tengan sobre esta evaluación, deben publicarla en el [Syllabus del curso](#) correspondiente a esta evaluación. También, sienta la libertad de responder dudas de tus pares si crees que conoces la respuesta. En caso de tener dudas que impliquen mostrar su solución o partes de ella, no utilicen este medio de consulta. Para estos casos, envíe un correo al cuerpo docente ([hfvaldivieso@uc.cl](mailto:hfvaldivieso@uc.cl)) o muestre su solución sólo en reunión personal cuando se reúna con algún miembro del cuerpo docente.

## 6. Anexo: Ejemplos de exámenes de semestres pasados

En los siguientes enlaces podrás encontrar las herramientas funcionales de varios exámenes del año 2020 al 2023 que siguieron instrucciones de evaluación similares al propuesto.

- [salón de la fama 2020.](#)
- [salón de la fama 2021.](#)
- [salón de la fama 2022.](#)
- [salón de la fama 2023-1.](#)
- [salón de la fama 2023-2.](#)

Son libres de revisar estos ejemplos como inspiración para encontrar su propio tema o *dataset* a trabajar. Además, es completamente posible usar el mismo tema o *dataset* que alguna de estas evaluaciones, ya que las tareas o contexto a resolver puede ser muy distinto. Pero también ojo que el **replicar o usar** como base una de estas evaluaciones (código o de diseño) caería como falta de integridad académica, como es indicado en el [programa del curso](#).