



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

Tarea 2

Implementación de herramientas de visualización

Fecha de inicio: **Viernes 23 de septiembre** a las **20:00:00** hrs

Fecha de entrega: **Viernes 7 de octubre** a las **20:00:00** hrs

Tiempo estimado de trabajo¹: **10 horas**

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.

Específicamente, esta evaluación se relaciona con los siguientes resultados de aprendizaje:

- **Construir herramientas de visualización interactivas mediante programación para resolver una necesidad de comunicación de información puntual.**
- **(Bonus) Evaluar herramientas de visualización según principios perceptivos, principios de diseño gráfico y la efectividad de sus componentes.**
- **(Bonus) Proponer modificaciones que mejoren la herramienta de visualización mediante uno o más principios que anteriormente no cumplía.**

El material del curso relevante para esta evaluación es el siguiente:

1. Introducción a HTML, CSS y SVG
2. Introducción a JavaScript y D3.js
3. Selecciones y *data join* en D3
4. Utilidades de D3 I y II
5. *Layouts* tabulares en D3

En esta evaluación se incluyeron una serie de *bonus* para poder optar a una mayor calificación. En la [Sección 7](#) se detalla las condiciones de estos *bonus*.

¹El tiempo estimado de trabajo es una aproximación realizada por el equipo docente del tiempo promedio que debería tomarle a una persona desarrollar esta evaluación. Esta toma el supuesto de que esta persona ha estudiado todos los contenidos del curso relacionados a esta evaluación con anticipación a la fecha de inicio de la evaluación. Al ser un promedio, no es una garantía que siempre se complete la evaluación antes de este tiempo. A su vez, esto no asume ni sugiere que son horas seguidas de trabajo, si no que es la suma de tiempo en dedicar a sus partes.

1. Implementación de herramientas de visualización

Esta evaluación busca que implementes, mediante programación y tecnologías web, una herramienta de visualización **artística** e interactiva para resolver una necesidad de comunicación de información puntual. Las tecnologías a utilizar en esta implementación son: HTML, CSS, SVG, JavaScript y D3.js.

Específicamente, se debe entregar como resultado dos documentos HTML, uno que introduzca la herramienta, y otro que produzca la herramienta de visualización artística. Esta herramienta deberá ser interactiva y basada en SVG para la situación puntual propuesta. El estilaje del documento se determina mediante declaraciones de CSS y funciona gracias a un programa escrito en JavaScript que utiliza la librería D3.js.

Dicha situación, contexto de ella, especificaciones de la herramienta y ejemplo de las visualizaciones se detallan en las siguientes secciones.

1.1. Museo Artístico

Gracias a la capacidad de D3.js de confeccionar visualizaciones desde el bajo nivel (dibujando cada figura y coordinando estas con los datos), las herramientas de visualización posibles son ilimitadas. Dentro de estas, las visualizaciones artísticas tienen un gran valor para poner a prueba las capacidades de D3.js y construir, de forma automática, visualizaciones diferentes a las que se pueden hacer con *software* de visualización.

La necesidad a satisfacer es la del director del Museo de Arte Morderno, Glenn D. Lowry. El director desea presentar un resumen sobre las diferentes categorías de obras que disponen y de los artistas que participan en cada categoría. Además, como director de un museo, un requisito base es presentar toda esta información de la forma más artística posible, dejando de lado algunos criterios o principios de visualización de información como la tasa de tinta, principio de expresividad, *lie factor*, entre otros.

La herramienta a crear será un portal del museo para dar a conocer la información de este y motive a las personas a conocer más sobre sus artistas. Por un lado, el director necesita que las personas puedan explorar el portal, por medio de una visualización, para conocer las diferentes categorías presentes en el museo y mostrarles un resumen de información básica de estas categorías como la cantidad de obras disponibles, la cantidad de artistas y la proporción de artistas masculinos y femeninos. Por otro lado, dada una categoría específica, el director busca presentar de una forma estética la información de los artistas que han contribuido a dicha categoría. Dentro de esta presentación se desea que las personas puedan identificar el estado del artista (vivo o fallecido), comparar sus edades, y ofrecer un resumen de la contribución del artista a la categoría elegida.

Los *datasets* a utilizar en esta evaluación son [CategoryProcessed.csv](#) y [ArtistProcessed.csv](#). Ambos fueron extraídos de Kaggle² y procesados³ para dejar una versión limpia y fácil de utilizar en la herramienta.

El primer *dataset* contiene la información de las distintas categorías presentes en el museo con una cantidad considerable de obras. El segundo, contiene la información de los diferentes artistas que han contribuido al museo.

Los archivos entregados son tales que es posible cargar ambos en un programa JavaScript y usarlos directamente para implementar la herramienta solicitada. No debes modificar estos archivos y cualquier procesamiento requerido, por ejemplo filtros o transformación de datos, debe hacerse en el código una vez cargado los *datasets*.

²<https://www.kaggle.com/datasets/momanyc/museum-collection>

³El procesamiento incluyó filtrado, reemplazo de valores nulos por -1, derivación de nuevos atributos, entre otros.

Cada fila de `CategoryProcessed.csv` representa una categoría y contiene las siguientes propiedades:

Propiedad	Descripción
Category	Nombre de la categoría. También conocido como “departamento” en el <i>dataset</i> original.
Artist	Cantidad de artistas que han contribuido a la categoría.
Artwork	Cantidad de obras en la categoría.
Male	Cantidad de artistas catalogados con género <i>Male</i> (masculino) en dicha categoría.
Female	Cantidad de artistas catalogados con género <i>Female</i> (femenino) en dicha categoría.

Cada fila de `ArtistProcessed.csv` representa un artista y contiene las siguientes propiedades:

Propiedad	Descripción
Artist	Nombre del artista.
Nacionality	Nacionalidad del artista.
Gender	Género del artista según la fuente original: <i>Male</i> (masculino) o <i>Female</i> (femenino).
BirthDay	Año de nacimiento.
DeathDay	Año de fallecimiento. Si es -1 , indica que el artista todavía no fallece.
TotalArtWork	Cantidad total de obras del artista en el Museo.
Categories	Diccionario con la cantidad de obras por categoría del Museo. ⁴

1.2. Herramienta objetivo, funcionalidades y requisitos

Como se mencionó anteriormente, la herramienta debe suplir varias necesidades y tareas para el usuario objetivo. Con esa intención, en esta sección se detallan las funcionalidades y requerimientos objetivo a implementar de forma explícita, para así orientar de mejor forma el desarrollo de tu herramienta.

Como adelanto, en la [Figura 1](#) y [Figura 2](#) se presentan ejemplos de los glifos a realizar en la primera y segunda visualización respectivamente. Ten claro que **no se espera una reproducción idéntica de estos ejemplos**, este es solo un referente para dar mayor claridad sobre los glifos a diseñar.

⁴Para procesar esta propiedad, se recomienda investigar [JSON.parse\(\)](#)

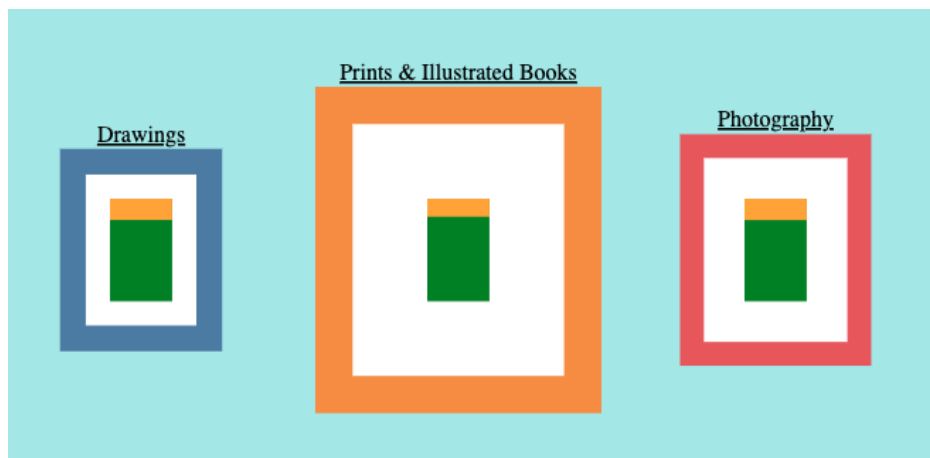


Figura 1: Ejemplo del glifo solicitado para la primera visualización: categorías del museo. La pintura del centro tiene un tamaño fijo y está compuesta por dos barras apiladas. El largo de estas es proporcional a la cantidad de artistas por género en dicha categoría. El grosor de la sección blanca (el paspartú) representa la cantidad de obras en dicha categoría. Finalmente, el grosor del marco representa la cantidad de artistas diferentes en dicha categoría.

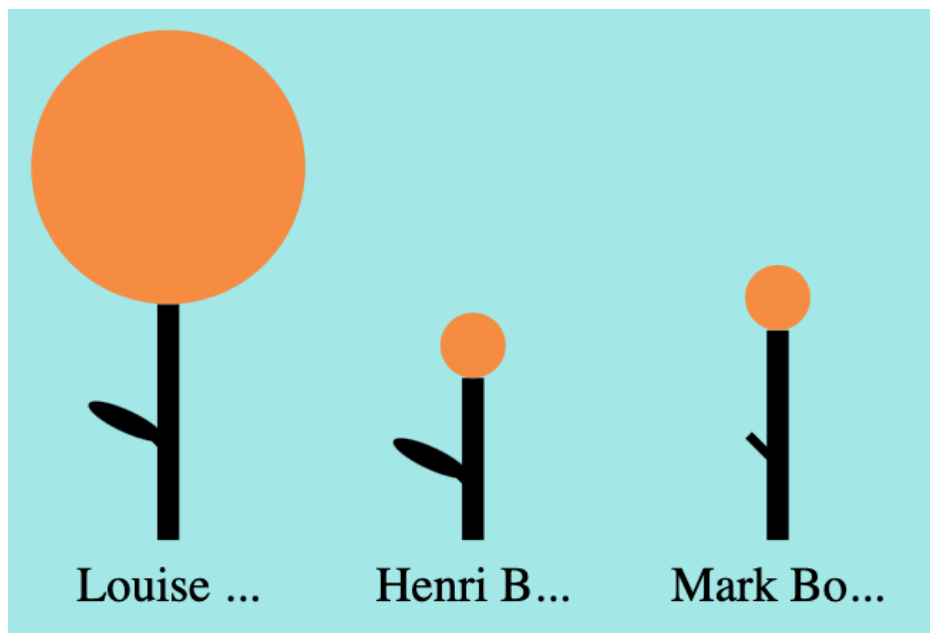


Figura 2: Ejemplo del glifo solicitado para la segunda visualización: artistas que contribuyeron a una categoría. El tamaño del círculo representa la cantidad de obras que ha creado cada artista en una categoría específica. El largo del tallo representa la edad, y la presencia de una hoja indica que dicho artista sigue vivo a la fecha.

A continuación se listan las funcionalidades y requisitos que se espera que cumpla la herramienta. Primero, aquellos generales de la herramienta:

- Construir un documento HTML que contenga información para contextualizar la herramienta. Los mínimos elementos que debe incluir este documento son: un título principal, algún tipo de descripción que entregue información del fondo de la visualización, el creador de la visualización (tu nombre completo), la fuente original de los datos utilizados (es decir [la página de Kaggle](#)). Final-

mente, agregar un botón o *link* que permita acceder a un segundo documento HTML.

- Construir un segundo documento HTML que contenga las dos visualizaciones solicitadas y **creadas a partir del data join de D3**.
- Para ambos documentos HTML, se debe utilizar estilaje con **external CSS** para cambiar el color de fondo y dejar un margen a los costados del documento. El color y la cantidad de margen quedan a tu criterio.
- Construir las visualizaciones a partir de los datos provistos (categorías y artistas). Aquellos archivos pueden ser cargados de forma local, es decir, archivos que viven en el mismo computador que ejecuta el programa, o mediante los siguientes enlaces: [CategoryProcessed.csv](#) y [ArtistProcessed.csv](#). Ambos archivos deben ser cargados con las función provista por la librería D3: `d3.csv()`.
- No debes utilizar `while`, `for` o `forEach` en el código. Para iterar los datos, se debe utilizar `.map()`, `.filter()`, **funciones de D3** y **data join**.
- La implementación no debe presentar errores de funcionamiento que interrumpan su experiencia de uso, ni errores de código durante su funcionamiento.

Segundo, las funcionalidades y requisitos relacionados a implementar la primera visualización, categorías del museo:

- Cada categoría será representada por un glifo de un cuadro de pintura. Este glifo se compone de:
 1. Un marco cuyo color es único por categoría y su grosor codifica la cantidad de artistas en dicha categoría. Debe utilizar las escalas de D3 para codificar el grosor.
 2. Una pintura, de tamaño fijo y dentro del marco, compuesta por dos barras apiladas, de diferente color, cuyos largos codifican la proporción de artistas masculinos y femeninos en dicha categoría.
 3. Un paspartú (separación entre la pintura y el marco) cuyo grosor codifica la cantidad de obras en dicha categoría. Debe utilizar las escalas de D3 para codificar el grosor.
 4. Un texto encima del cuadro con el nombre de la categoría. Este texto debe estar subrayado.
- Los glifos no se pueden superponer.
- Solamente cuando el cursor (*mouse*) esté sobre el glifo de las categorías, se debe desplegar, como un texto, los porcentajes exactos de la proporción de artistas masculinos y femeninos. Queda a tu criterio dónde se muestra dicha información.
- Es posible presionar, con el cursor (*mouse*), un glifo para actualizar la segunda visualización. Esta actualización permitirá mostrar únicamente a los artistas con obras pertenecientes a la categoría que representa el glifo. Además, deberás dejar una marca visual que indique que dicho glifo fue seleccionado. Por ejemplo, reducir la opacidad de los demás glifos, agregar un marco adicional al seleccionado, entre otros.

Tercero, las funcionalidades y requisitos relacionados a implementar la visualización de los artistas.

- Cada artista será representado por un glifo de una flor. Este glifo se compone de:

1. Un tallo cuyo largo codifica la edad del artista a la fecha actual. En caso que el artista ya haya fallecido, se codifica la edad a la fecha de fallecimiento.
 2. Una bifurcación junto al tallo para codificar el estado del artista. Si el artista sigue vivo, la bifurcación debe incluir una hoja del mismo color del tallo. En caso de haber fallecido, la bifurcación no debe contener la hoja, es decir, solo se verá la bifurcación. En la **Figura 2** se pueden observar ejemplos para dos artistas que siguen vivos y otro que falleció.
 3. El pétalo de la flor, representado por un círculo, cuyo tamaño representa la contribución de obras del artista a la categoría seleccionada. El color del pétalo debe coincidir con el color del marco de la categoría correspondiente.
 4. Un texto debajo del árbol con el nombre del artista. Este debe contener máximo 10 caracteres. Por lo tanto, si el nombre tiene 11 o más caracteres, debes tomar los 7 primeros y agregar “...” al final.
- Los glifos no se pueden superponer.
 - Máximo se deben mostrar 100 artistas en esta visualización. Para esto, debes seleccionar **aleatoriamente** 100 artistas del *dataset* que tengan obras en la categoría seleccionada. Para la selección aleatoria y acotada, puedes hacer:

```
artistas_filtrados.filter(d => Math.random() > 0.7).slice(0, 100)
```

- Solamente cuando el cursor (*mouse*) esté sobre un glifo del artista, este se debe destacar sobre los demás. Para lograr esto, una opción es reducir la opacidad de los demás glifos a excepción del que deseamos destacar. Además, se debe desplegar un *tooltip*, junto a la posición del *cursor*, que muestre la información básica del artista destacado: nombre completo, género, nacionalidad, fecha de nacimiento y edad.
- Incluir al menos **un filtro** para reducir la cantidad de artistas en pantalla. Este filtro debe estar vinculado a algún atributo del artista. El filtro específico a crear queda a tu criterio, algunos ejemplos son mostrar artistas con: un género específico, fallecidos, con cierto rango de edad, con una cantidad mínima de obras, entre otros. Además, debes incluir una forma de revertir este filtro para ver todos los artistas mostrados inicialmente. Esto puede ser mediante un botón de “reinicio”.
- Contener un **input de selección** que contenga las siguientes opciones: orden según el *dataset*, ordenar por nombre alfabético u ordenar por edad.
- Al cambiar el valor del *input* de selección, se reordenan los glifos según la opción elegida. Este orden debe darse mediante una animación gradual en el tiempo. Queda a tu criterio si el orden es ascendente o descendente.
- Utilizar animaciones en el tiempo (transiciones de D3) cuando se crea, actualiza o filtra la visualización de los artistas. Recuerde el **data join de D3.js**: *enter*, *update* y *exit*. La animación en específico a crear para cada elemento del glifo queda a tu criterio.

1.3. Libertades y mínimos de implementación

Como fue mencionado previamente, no se espera un resultado idéntico visualmente al ejemplo provisto, mientras las funcionalidades implementadas sean las pedidas y se respete las marcas y canales mencionados por glifo.

Te invitamos a explorar una paleta de colores distinta para el fondo, texto y elementos de vistas. Además, puedes probar con distintos tamaños y fuentes, pero procura que sea legible y haga sentido.

Si hay algún aspecto a implementar que no te queda claro si permite libertad de realización, por favor consúltalo en el [canal de Discord](#) correspondiente a esta evaluación.

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.
- Puedes hacer uso tanto de la versión 6 o 7 de D3.js, pero no una anterior a ambas.
- 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.

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 el resultado de la herramienta entregada y se usará una rúbrica como guía. Esta detalla niveles de cumplimiento para los distintos criterios establecidos y solicitados y puedes encontrarla [en el siguiente spreadsheet](#) (en la hoja llamada Tarea 2).

Además de determinar el nivel de desempeño alcanzado, el equipo docente adjuntará retroalimentación escrita que complemente la información de la rúbrica.

Cada criterio tiene cierta ponderación de puntaje. El multiplicar el nivel obtenido en el criterio por su ponderador correspondiente es equivalente al puntaje obtenido en dicho criterio. La nota en la evaluación se obtiene de comparar la suma de puntajes obtenidos en todos los criterios y el máximo posible al lograr el mayor desempeño en todos los criterios. Se usará una escala lineal, donde el puntaje mínimo obtiene nota 1, y el máximo 7.

3. Entregables

Se espera que el entregable corresponda a: **una herramienta de visualización artística de información**. Este resultado debe entregarse como dos documentos **HTML**, extendidos mediante archivos **CSS** y **JavaScript**. No se aceptarán entregas en cualquier otro formato distinto al indicado anteriormente (archivos PDF, TXT, DOC, etc.). De no entregar o entregar un formato diferente al especificado, no se revisará la entrega y se colocará nota mínima.

La entrega se realizará mediante un archivo comprimido de extensión **ZIP** que contenga la herramienta, mediante la plataforma Canvas en la [evaluación](#) correspondiente. Cualquier otro archivo (imágenes, hojas de estilo, *scripts*, etc.) que sea necesario para la visualización correcta de tu entrega debe también ir incluido en el archivo comprimido. Finalmente, no es necesario que se incluyan los archivos de datos en la entrega.

4. Dudas

Cualquier duda que tengas sobre esta evaluación, prefiere publicarla en el [canal de Discord](#) 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 los canales públicos del Discord. Para estos casos, envíe un correo al cuerpo docente o muestre su código 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 **4 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 atrasadas, la nota final de la tarea se calcula mediante la siguiente fórmula:

$$\text{mín}((7 - (0,5 \times \text{días_atraso})), \text{nota_obtenida})$$

Por otro lado, entregas con más de 4 días (96 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, 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.

De preferencia escribe un correo a la ayudante de bienestar (tpcarrasco@uc.cl) o al docente del curso (hfvaldivieso@uc.cl).

7. **Bonus de la evaluación (Décimas o puntaje recuperativo RC)**

Esta evaluación presenta una serie de *bonus* para recuperar un puntaje de Revisión de Contenidos (RC) o 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, a excepción del *bonus* para el puntaje RC o para la autoevaluación, **todos los demás *bonus*** requieren que la nota en tu tarea (sin ningún *bonus*) debe ser **igual o superior a 4.0**.

Finalmente, para esta evaluación, se puede obtener un máximo de 10 décimas como bonificación. Además, en caso de entregas atrasadas, las bonificaciones **no podrán sobrepasar la nota definida por el descuento por atraso**.

7.1. **Ejemplificación de Categoría (1 punto RC bonus)**

Se otorgará un punto RC *bonus* si se implementan 2 botones en la primera visualización: “Ver ejemplos” y “Ocultar ejemplos”. El primer botón se encargará de reemplazar cada pintura de los cuadros por una imagen de una obra representativa para dicha categoría. Para lograr esto, deberás investigar por el [tag <image> en SVG](#). El segundo botón permite ocultar las imágenes para volver a ver la proporción de artistas en cada categoría.

7.2. **Crítica de visualización y propuesta de mejora (3 décimas)**

Se otorgará una bonificación de 3 décimas si en el primer documento HTML se incluye una crítica negativa por visualización y una mejora para solucionar esa crítica. Dado que esta evaluación consiste en dos visualizaciones, se espera en total dos críticas y dos mejoras. Finalmente, estas críticas deben estar justificadas en el principio de expresividad, de efectividad, de diseño o de percepción.

7.3. **Documento Responsive (2 décimas)**

Se otorgará una bonificación de 2 décimas si ambos documentos HTML son *responsive*, es decir, se ajustan al tamaño de la ventana. Este requerimiento incluye la visualización. Se recomienda investigar sobre [viewport](#) en SVG. En [Google](#) encontrarán mucha información sobre este concepto.

7.4. **Marcos realistas (2 décimas)**

Se otorgará una bonificación de 2 décimas si se mejora el diseño de los marcos con ayuda **únicamente de CSS y SVG**. Esta mejora de diseño implica cambiar el marco rectangular por un diseño más artístico. Por ejemplo, agregar decoración o mejorar las esquinas con más elementos para simular marcos más realistas como los de [esta imagen](#). Además, esta mejora de diseño no debe distraer o eliminar la codificación solicitada de ancho y color.

7.5. **Codificación del género del artista (1 décimas)**

Se otorgará una bonificación de 1 décima si se incluye alguna marca o canal en cada glifo de artista que permita identificar el género de este. Por ejemplo, aumentar la cantidad de ramas y hojas según el género, incluir un pasto, utilizar 2 tallos, etc. El único requisito para este *bonus* es que la marca y/o canal utilizado para codificar el género no debe interferir con la otra información codificada.

7.6. Codificación de la contribución a diversas categorías (4 décimas)

Se otorgará una bonificación de 4 décimas si se reemplaza el pétalo del glifo del artista por un conjunto de pétalos concéntricos. Cada círculo representa una categoría distinta donde el artista ha contribuido con obras. El color debe coincidir con el color del marco de la categoría correspondiente. El tamaño del círculo codifica la cantidad de obras de dicha categoría. Los pétalos deben estar ordenados, el más grande debe estar al fondo, mientras el más chico al frente. En la **Figura 3** podrás ver un ejemplo de este *bonus*.⁵

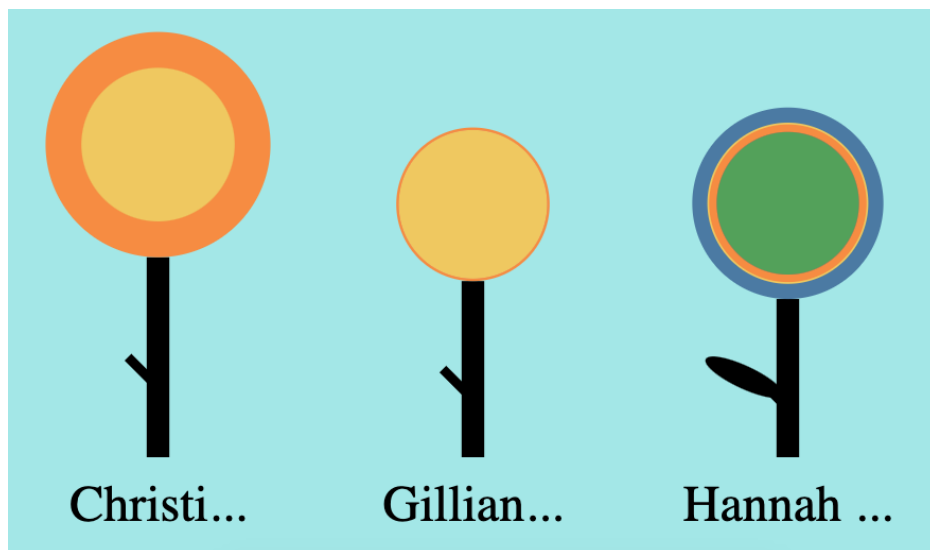


Figura 3: Ejemplo del glifo *bonus* solicitado para la segunda visualización: artistas que contribuyeron a cada categoría. Ahora, para cada artista se incluyen pétalos concéntricos para codificar la contribución del artista a cada categoría posible

7.7. Autoevaluación (Bonus de 2 décimas)

Luego de finalizar el plazo total de entrega (considerando los 4 días de atraso), se habilitará un formulario que te permitirá realizar la autoevaluación de tu tarea. Este ejercicio será **totalmente voluntario** y no afectará negativamente tu nota. Luego, se comparará la nota de tu autoevaluación con la nota entregada por el cuerpo docente. Si la diferencia absoluta entre ambas notas es menor o igual a tres décimas, recibirás una bonificación de dos décimas en la nota final de tu tarea. Para la comparación de ambas notas, se utilizará la nota final después del periodo de re-corrección. Esta considerará el puntaje obtenido en la rúbrica junto con los *bonus* y el descuento por atraso.

A modo de ejemplo, si la nota entregada por el cuerpo docente (después de todo el proceso de re-corrección, considerando los *bonus* y el descuento por atraso) fue de un 6.00 y la nota de autoevaluación está entre un 5.70 y 6.30, la tarea tendrá la bonificación de 2 décimas y la nota final será un 6.20. En otro caso que la autoevaluación fuera mayor a 6.30 o menor a 5.70, no obtendrás el *bonus*.

⁵*Hint*: para lograr esto, primero deberás hacer un procesamiento a la columna `Categories`. Se recomienda estudiar `Object.keys()`, `Object.entries()` y `Object.values()` para transformar un diccionario en una lista y poder utilizar `.map`, `.filter`, `data join`, etc.