



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

## Tarea 3

### D3.js avanzado

Fecha de inicio: **Miércoles 12 de abril** a las **20:00:00** hrs

Fecha de entrega: **Jueves 27 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. 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 únicamente 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.

Además, un problema común en el desarrollo de visualizaciones es continuar trabajo heredado de otras personas sin empezar desde cero todo el trabajo. Por lo cual, en esta tarea te enfrentarás a tener que completar un código que ya viene con ciertos elementos y lógicas programados, y que no podrás modificar.

## 1.1. IMDB Movies

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.

Para esta ocasión, buscamos reflejar de una forma más “elegante”, información de diversas películas de IMDB mediante una página web. Por un lado, queremos que las personas puedan explorar esta página, por medio de una visualización, para conocer 3 principales géneros de películas (comedia, romance y acción) y mostrarles un resumen de información básica de estas categorías como la cantidad de películas por género y la cantidad promedio de votos que le han dado a películas de dicho género. Por otro lado, dado un género específico, queremos presentar de una forma estética la información de las películas que presentan dicho género. Dentro de esta presentación se desea que las personas puedan identificar el *rating* de la película (mala, promedio o buena) y ofrecer un resumen sobre el tipo de película y la duración de esta.

Para esta evaluación se utilizarán dos *datasets*. Uno generado artificialmente en el mismo código JS entregado y el otro que se encuentra en [este link](#). Este último fue extraído de IMDB<sup>1</sup> y procesado<sup>2</sup> para dejar una versión limpia y fácil de utilizar en la herramienta.

El primer *dataset*, el cual es generado de forma artificial en el código, contiene la información de los distintos géneros de película que vamos a visualizar en la tarea: comedia, romance y acción. El segundo, contiene la información de las diferentes películas que presentan alguno de los géneros mencionados anteriormente. No debes modificar el *dataset* base y cualquier procesamiento requerido, por ejemplo filtros o transformación de datos, debe hacerse en el código una vez cargado el *dataset*.

Cada elemento del *dataset* artificial representa un género y contiene las siguientes propiedades:

Propiedad	Descripción
Genre	Nombre del género.
MoviesCount	Cantidad de películas que contienen dicho género.
MoviesVoteAverage	Cantidad de votos en las películas que contienen dicho género.
AverageRating	Puntaje promedio de las películas que contienen dicho género.
AverageDuration	Duración promedio de las películas que contienen dicho género.

Cada elemento de `movies.json` representa una película y contiene las siguientes propiedades:

Propiedad	Descripción
Title	Nombre de la película. Este es único por película.
Type	Tipo de película: <i>tvMovie</i> , <i>videoGame</i> , <i>movie</i> y <i>short</i> .
Release_Year	Año de estreno.
Duration	Duración en minutos.
Genres	Lista de géneros de la película.
Category_Rating	Clasificación de la película según su puntaje: <i>Good</i> , <i>Average</i> o <i>Bad</i> .
Numeric_Rating	Puntaje numérico de la película.

<sup>1</sup><https://www.imdb.com/interfaces/>

<sup>2</sup>El procesamiento incluyó filtrado, derivación de nuevos atributos, entre otros.

## 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 las figuras artísticas 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 las visualizaciones a diseñar.

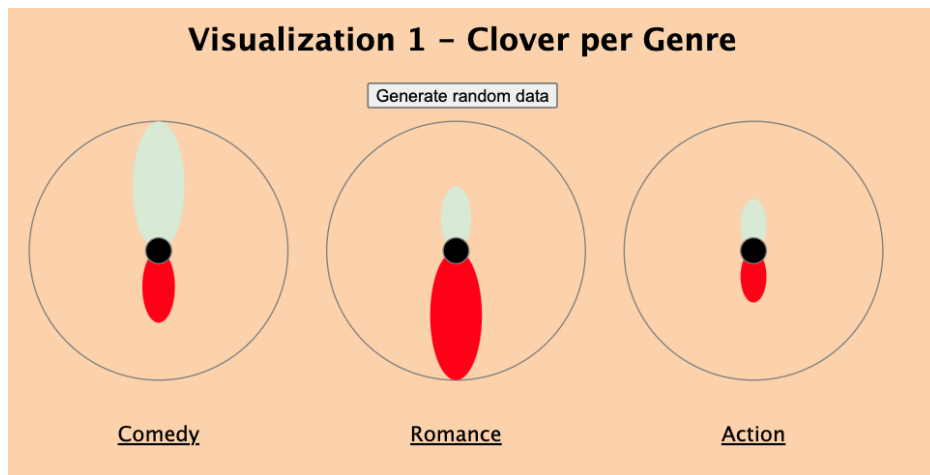


Figura 1: Ejemplo de la primera visualización: géneros de películas. Cada trebol representa un género y sus hoja representan una característica del género. Su tamaño y posición es proporcional a la información que codifica. Una hoja codifica la **cantidad de películas** y la otra codifica la **cantidad de votos promedio** en dicho género. Cada hoja dentro del pétalo es de un color distinto según la característica que definen. Finalmente se incluye un círculo central de color negro en el trébol solo con fines estéticos, y un círculo externo para indicar el tamaño máximo del trébol.

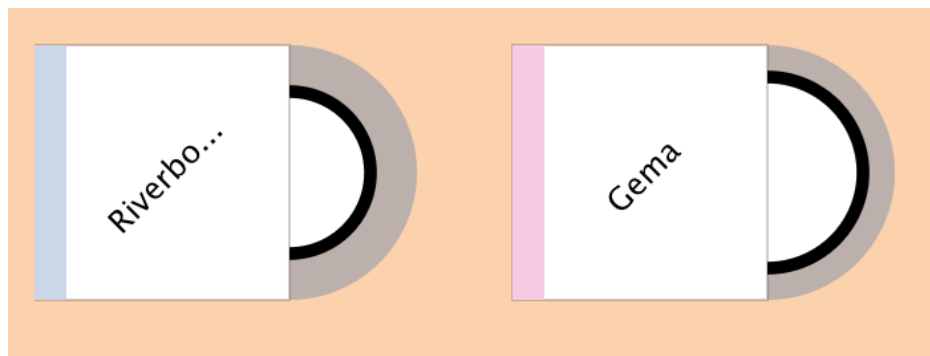


Figura 2: Ejemplo de la segunda visualización: películas que presentan un género específico. Cada caja con su disco representa una película. El color del extremo izquierdo representa el tipo de película, el color del disco representa el *rating* de la película (mala, buena o promedio), y el tamaño de área de grabación del disco representa la duración de la película.

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 para presentar la herramienta. Los mínimos elementos que debe incluir este documento son: (1) un título principal, (2) una descripción de máximo 200 palabras que entregue información sobre las visualizaciones a observar, (3) el creador de la visualización (tu nombre completo), (4) citar la fuente original de los datos utilizados (es decir [la página de IMDB](#)), y (5) un botón o *link* que permita acceder a `vis.html`. Finalmente, debes incluir 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.
- Utilizar el archivo `vis.html` y `main.js` para construir las dos visualizaciones solicitadas y **creadas a partir del `data join de D3`**.
- 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, géneros de películas:

- Cada género será representado por un trébol. Estos no deben superponerse y se componen de:
  1. Una hoja con forma de elipse apuntando hacia arriba cuyo tamaño (ancho y largo) codifica proporcionalmente *MoviesCount*. El color a utilizar queda a tu criterio.
  2. Una hoja con forma de elipse apuntando hacia abajo cuyo tamaño (ancho y largo) codifica *MoviesVoteAverage*. El color a utilizar debe ser diferente a la primera hoja.
  3. Un pequeño círculo en el centro del trébol de tamaño fijo, color negro y sobre las hojas. Ambas hojas deben tocar este pequeño círculo.
  4. Un círculo externo de color transparente y orilla gris para encerrar el trébol. Su tamaño es tal que la hoja más grande posible de entre todos los tréboles va a tocar justo este círculo.
  5. Un texto debajo del trébol con el nombre del género. Este texto debe estar subrayado.
- Para cada tipo de hoja (hoja para arriba y hoja para abajo) se debe utilizar escalas de D3 independientes, es decir, una escala para trabajar el atributo *MoviesCount* y otra escala para el atributo *MoviesVoteAverage*. Además, debes ajustar el dominio de cada escala para que el valor más chico de los 3 géneros sea la hoja más pequeña posible y el valor más grande de los 3 géneros sea la hoja más grande posible. Se recomienda investigar el método `extent` de D3.
- Solo cuando el cursor (*mouse*) esté sobre el trébol, se debe desplegar, como un texto, todos los atributos del género asociado al trébol. Unz vez retirado el cursos, la información se deja de observar. Queda a tu criterio el formato para mostrar dicha información.
- Es posible presionar, con el cursor (*mouse*), un trébol para actualizar la segunda visualización. Esta actualización permitirá mostrar únicamente las películas que entre sus géneros esté incluido el que representa el trébol. Además, deberás dejar una marca visual, en esta visualización, que indique que dicho trébol fue seleccionado.
- Se incluye un botón para actualizar los datos de esta visualización. se deberá utilizar animaciones en el tiempo (transiciones de D3) para actualiza el tamaño de las hojas. Recuerda el **`data join de D3.js`**: `enter`, `update` y `exit`.

- Se incluyen 3 botones que simulan el presionar cada uno de los tréboles. Una vez presionado uno de estos botones, se debe destacar el trébol correspondiente al género presionado.

Tercero, las funcionalidades y requisitos relacionados a la segunda visualización: las películas.

- Cada película será representada por una caja de DVD con un disco a medio salir. Estos DVD con su caja no se pueden superponer y se componen de:
  1. Un extremo de ancho fijo en la caja cuyo color representa el tipo de película. Cada tipo debe tener un color distinto.
  2. Una caja de color blanco y borde gris con el título de la película. Este título debe estar rotado en forma diagonal y debe contener máximo 10 caracteres. Por lo tanto, si el nombre tiene 11 o más caracteres, debes tomar los 7 primeros y agregar 3 puntos ("...") al final del fragmento tomado. De este modo, los 7 caracteres + los 3 puntos harán los 10 caracteres totales.
  3. Un disco (DVD) a medio salir de la caja cuyo color representa el *rating* de la película. Debe ser un color distinto para cada valor posible: "mala", "buena" o "promedio". Además, el tamaño del área de grabación del disco debe ser proporcional a la duración de la película, mientras más grande es esa área, la película dura más. El tamaño total del disco es fijo en toda la visualización.
  4. Un círculo central del DVD de color blanco para representar el *HUB*. Este círculo debe incluir un borde negro.
- Se deben mostrar entre 20 y 30 películas en esta visualización y disponer entre 4 a 6 películas por fila, tu decides cuantas poner por fila mientras respetes la cantidad<sup>3</sup>. Para la selección acotada se recomienda investigar el método `slice` de una lista.
- Solo mientras el *cursor* esté sobre una de las películas, se debe reducir la opacidad de todas las demás a excepción de la que tiene el *mouse* encima. Además, se debe desplegar un *tooltip*, junto a la posición del *cursor*, que muestre la información básica de la película: nombre, cantidad de géneros, tipo, duración, fecha de estreno, *rating* y *rating* numérico. Una vez retirado el cursos, el *tooltip* se debe dejar de observar.
- Poder filtrar las películas por su *rating* numérico. Además de poder revertir este filtro para ver todas las películas mostradas inicialmente. Para esto, se incluyen dos botones en el HTML con toda la lógica programada para que llamen a la función encargada de crear la segunda visualización. Solo falta la parte de filtrar el *dataset*.
- Poder cambiar el orden de las películas según: su título, la fecha de estreno o el orden por defecto (el orden original con el que vienen en el *dataset*), Este re-orden de DVD debe darse mediante una animación gradual en el tiempo. Queda a tu criterio si el orden es ascendente o descendente. Toda la lógica del selector ya está programada, solo falta reordenar el *dataset* cuando corresponda y aplicar la animación.
- Utilizar animaciones en el tiempo (transiciones de D3) cuando se crea, actualiza y filtra la visualización de las películas. Recuerde el `data join de D3.js`: *enter*, *update* y *exit*. Las animaciones en específico a utilizar quedan a tu criterio.

---

<sup>3</sup>Se recomienda investigar sobre módulo y división entera en Javascript.

### 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 el diseño de cada figura (trébol y caja con DVD).

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 [foro del Syllabus](#).

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.
- 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.
- Las visualizaciones deben ser desarrolladas con elementos propios de un SVG (`rect`, `circle`, `line`, `text`, `ellipse`, `g`, `path`).
- Cada visualización debe estar contenida en un único SVG. Por lo tanto, `vis.html` solo puede contener 2 SVG.
- 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.**

### 1.4. Archivos entregados

Para esta tarea, se te hará entrega de algunos archivos con varios elementos ya programados. Su trabajo será completar los archivos con lo solicitado en esta evaluación:

- `vis.html`: archivo HTML con los elementos necesarios para confeccionar las visualizaciones. **No debes modificar este archivo.**
- `style.css`: archivo CSS que aplica estilo a `vis.html`. Puedes modificar su contenido, pero no el nombre del archivo.
- `configuration.js`: archivo JS encargado de: (1) preparar toda la lógica de los datos artificiales de la primera visualización, y (2) configurar la lógica de los botones y selectores.

Se incluye una función llamada `preprocessingMoviesDataset` que recibe el género seleccionado y si hay o no que filtrar los datos. Esta función llamará a `createDVDs` encargada de confeccionar la segunda visualización **No debes modificar nada de este archivo.**

- `main.js`: archivo JS con 2 funciones que deberás completar para el correcto desarrollo de la evaluación.

1. `createClover(dataset)`

Esta función se llama cada vez que se crea o actualiza la primera visualización. `dataset` es una lista de diccionario con la información de los 3 géneros a visualizar como trébol. **Importante:** Cada vez que se haga *click* en un trébol, deberás llamar a `preprocessingMoviesDataset` con la información del género seleccionado e indicando que no hay que filtrar. Esta función se encargará, posteriormente, de llamar a `createDVDs`.

2. `createDVDs(dataset, genre, filter_dataset)`

Esta función se llama cada vez que se crea o actualiza la segunda visualización. `dataset` es una lista de diccionario con la información de todas las películas, `genre` es un *string* con el género seleccionado a visualizar y `filter_dataset` es un booleano indicando si hay que filtrar o no los datos según el *rating* numérico.

## 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. La mayoría de los ítems serán calificados de forma ternaria: cumple totalmente el ítem, cumple parcialmente el ítem, o no cumple el ítem, aunque hay algunos que por su naturaleza binaria, serán calificados como cumple o no cumple. 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 3".

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: **una herramienta de visualización artística de información a partir de código JS y D3.js**. Este resultado debe entregarse como un ZIP como mínimo cinco archivos: un documento **HTML** para la primera página, el documento **HTML** entregado para las visualización, los dos archivos **JS** y el archivo de estilo **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. **No debes incluir el enunciado y el dataset en el ZIP.**

## 4. Dudas

Cualquier duda que tengas sobre esta evaluación, prefiere publicarla en el [Syllabus del curso](#). También, sienta 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 atrasadas, la nota final de la tarea se calcula mediante la siguiente fórmula:

$$\text{mín}(7 - (0.5 \times \text{días\_atraso}), \text{nota\_obtenida}) + \text{bonus}$$

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 esta flexibilidad posterior a los 3 días de atraso, escribir al docente del curso ([hvaldivieso@ing.puc.cl](mailto: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**.

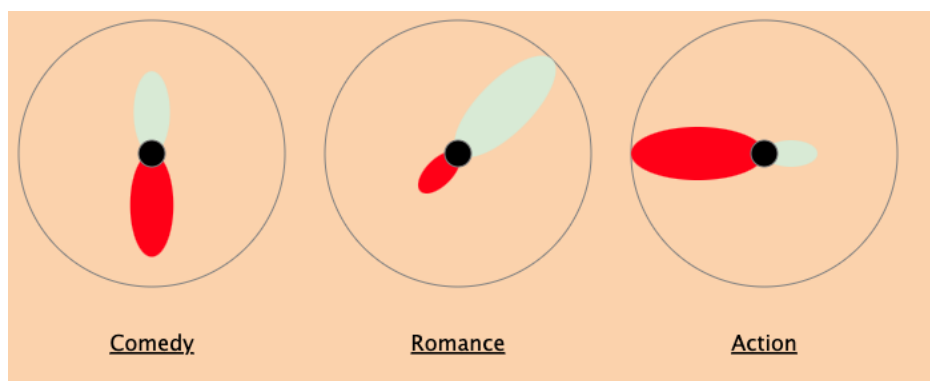
Finalmente, las bonificaciones **se aplican después de la nota definida por el descuento por atraso**.

### 7.1. Rotación de trébol (1 décimas)

Se otorgará una bonificación de 1 décima si los tréboles de la primera visualización presentan la siguiente disposición:

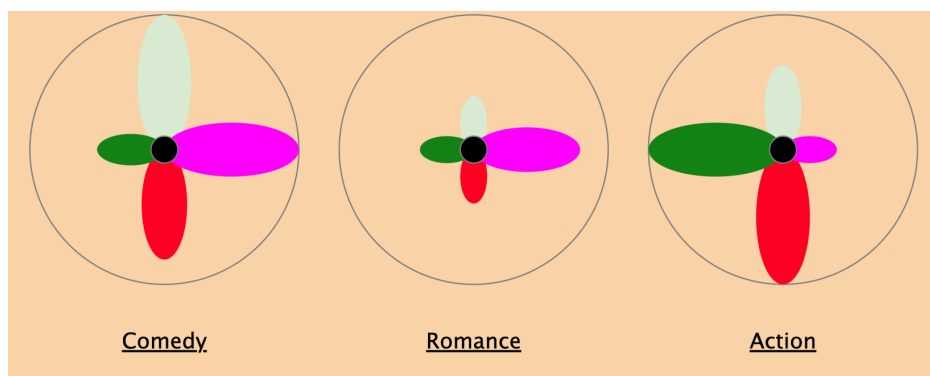
En otras palabras, el primer trébol mantiene la disposición vertical como en la visualización original, pero el segunda tiene las hojas en diagonal, y en el tercero las hojas están de forma horizontal.





### 7.2. Más hojas en el trebol (2 décimas)

Se otorgará una bonificación de 2 décimas si en vez de mostrar 2 hojas por trébol, se muestran 4 hojas para incluir la información de *rating-promedio* y *duracion-promedio*. Estas 2 nuevas hojas deben estar perpendicular a las hojas creadas originalmente. A continuación se muestra un ejemplo de las tréboles con estas 4 hojas.



### 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. **Importante:** debes hacer esto solo utilizando comandos de CSS y JS, y por lo tanto, no puedes modificar `vis.html` bajo ningún motivo.