

# Tarea Grande 1

Visualización de Información

Ayudantes: Constanza Olate, Daniela Poblete y Ricardo Schilling.

Profesor: Denis Parra

Enunciado: 2 de septiembre de 2020

---

## Indicaciones

- Fecha de entrega: 16 de septiembre de 2020 hasta las 23:59.
  - Se debe entregar la tarea en el repositorio asignado a cada uno por *Github Classroom*.
  - La tarea es **en parejas**.
  - **La copia será sancionada con una nota 1.1 en la tarea, además de las sanciones disciplinarias correspondientes.**
- 

## Objetivo

El objetivo de esta tarea es que aprendan a:

- Utilizar **Python** para manipular datos.
- Utilizar datos para hacer gráficos utilizando la **librería Altair**.
- Aprender nociones básicas de la utilidad de una **API**.
- Utilizar **HTML+CSS+JS** para construir una página web sencilla en la que se muestren las visualizaciones construidas.

# Contexto

## Reglas Generales

En la parte 1 de su tarea trabajarán con los datos entregados, con la finalidad de obtener lo necesario para la confección de un *dashboard*.

Para esto es importante que consideren lo siguiente:

- **El entregable de la sección 1 es un *jupyter notebook* (extensión.ipynb)**, donde se encuentre el **código, gráficos realizados y explicaciones** que les parezcan pertinentes.
- Se deberá utilizar **obligatoriamente** la librería ***Pandas*** para el procesamiento de los datos.
- De no cumplir con alguno de los dos puntos anteriores, su nota máxima será un 4.
- Se espera que usen la librería *Pandas* de manera **eficiente** (por ejemplo, **evitando loops innecesarios**). De lo contrario, se aplicarán descuentos en su nota. Lo anterior es a criterio del ayudante.
- Los **gráficos deben realizarse obligatoriamente con la librería *Altair***. Si no utilizan la librería anteriormente mencionada para las visualizaciones, **no obtendrán puntaje en los ítems relacionados con este punto**.
- Los gráficos deben tener título y nombres para cada eje según corresponda, sino se aplicará un descuento por cada gráfico que no cumpla con esto.

## Datos entregados

En el archivo .csv adjunto a esta tarea encontrarán la siguiente información de películas:

- **ID**: ID del dataset.
- **Title**: El título de la película.
- **Year**: Año de producción.
- **Age**: *Target audience* o Grupo objetivo.
- **IMDb**: *IMDb rating*.
- **Rotten Tomatoes**: *Rotten Tomatoes rating, Tomatometer®*.

- **Netflix:** Si se encuentra en la plataforma = 1 si no = 0.
- **Prime Video:** Si se encuentra en la plataforma = 1 si no = 0.
- **Disney+:** Si se encuentra en la plataforma = 1 si no = 0.
- **Type:** película = 0, serieTV = 1.
- **Directors:** El/los director/es de la película.
- **Genres:** Género de la película.
- **Country:** País de origen.
- **Language:** Lenguajes disponibles.
- **Runtime:** Duración de la película en minutos.

## Instrucciones

### Parte 1: Manipulación y visualización de datos.

#### Pregunta 1

Queremos analizar la distribución típica de los puntajes promedio de las películas (para ver, por ejemplo, qué nota tienden a poner los usuarios). Para esto, deben realizar diez **gráficos de barras** cinco para *IMDb* y otros cinco para *Rotten Tomatoes*. Esto quiere decir que en su eje x tenga las notas de 1 a 10 y el otro de 0 a 100 %, y que en el eje y tengan la cantidad de películas que tienen esa nota.

Se espera que dentro de los cinco gráficos, se muestre uno con todas las películas, otro sólo con las películas que existen en *Netflix*, *Hulu*, *Prime Video* y *Disney+*.

Para realizar los gráficos de *IMDb*, deben redondear el puntaje de cada película para llevarlo a un número entero (una película con *rating* 1.2 debería dibujarse en la barra 1, y una película con *rating* 5.5 en la barra 6).

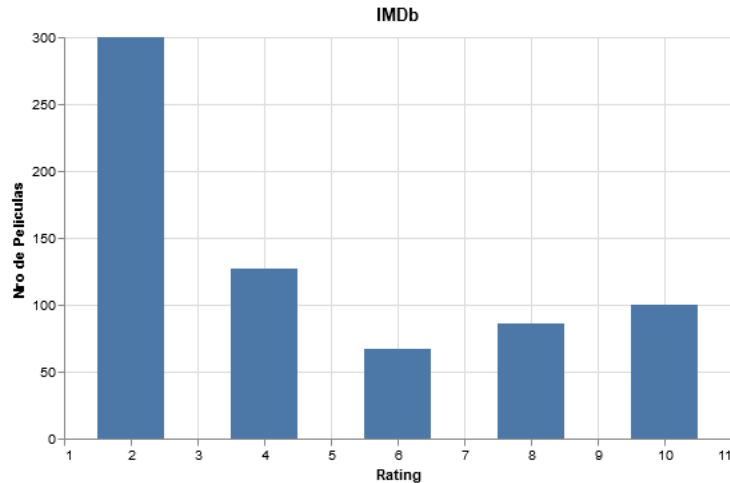


Figura 1: Ejemplo del gráfico de barras.

## Pregunta 2

Adicionalmente, queremos maximizar el tiempo que tenemos para ver películas... Así que, ¿Qué tal si buscamos las que tienen mejores notas?

Deben realizar un gráfico de dispersión de películas según su puntaje. En su eje  $x$ , deben estar los puntajes según *rotten tomatoes*, y en el eje  $y$  debe estar el puntaje de *IMDb*.

Adicionalmente, queremos ver rápidamente cuál veremos primero y por cuáles seguiremos.

En este gráfico, el color de cada punto debe representar el idioma de una película (solo es necesario representar 1 idioma por película) y el tamaño de cada punto debe representar el *runtime* de esta.

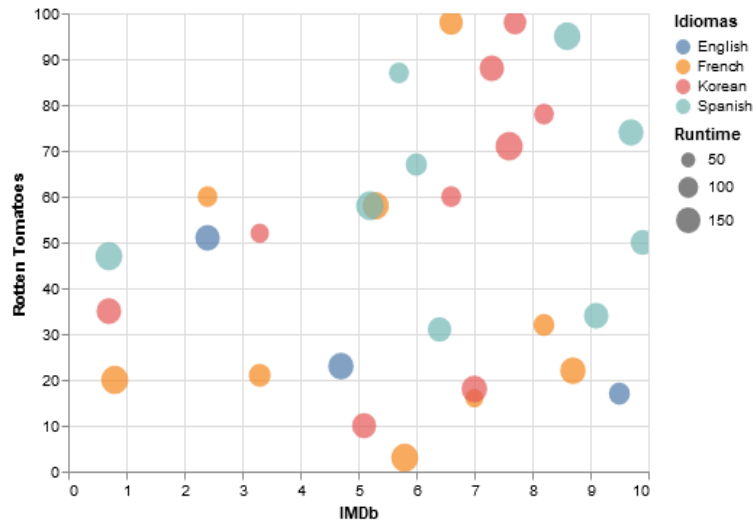


Figura 2: Ejemplo del gráfico de dispersión.

### Pregunta 3

También sería interesante saber si el promedio del puntaje de las películas se relaciona con el año en el que fue lanzado.

Para esto deben realizar un gráfico de barras en el que los años se encuentren en el eje x y el promedio de las películas lanzadas durante ese año se encuentren en el eje y.

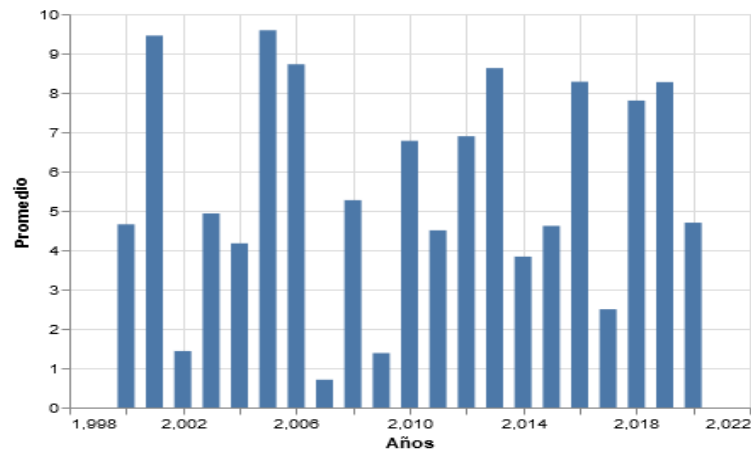


Figura 3: Ejemplo del gráfico de barras.

## Parte 2: *Dashboard* (2.5 puntos)

Con las visualizaciones ya generadas, deben construir una página *web* que contenga cada uno de los gráficos mencionados en el enunciado.

Se recomienda revisar la siguiente guía de como funciona *Flexbox*.

Como la confección de los gráficos será evaluada en la parte 1, en esta parte se espera que cumplan con lo siguiente:

1. Lista de gráficos: La página debe mostrar una lista de tres gráficos. Los cuales deben estar centrados y alineados.

A continuación se detallan los requisitos de cada elemento *HTML* en la página:

- **El cuerpo:**

- Debe tener un margen *responsive* que toma el 5 % de la pantalla.
- Debe ser un contenedor *flex*.
- Debe tener el color hexadecimal #fafbfc de fondo.

- **El encabezado:**

- Debe tener una altura de 50px.
- Debe ser un contenedor *flex*.
- Los elementos dentro deben estar **centrados**.

- **Los botones:**

- Debe tener los bordes redondos con 7px.

- **El contenedor que contiene a todas las imágenes:**

- Debe ser un contenedor *flex*.
- Los elementos dentro deben estar **centrados**.
- Los elementos dentro deben saltar de línea cuando no hay suficiente espacio (Ayuda: ver las propiedades de los contenedores *flex* en la guía de *Flexbox* y anexo del enunciado).

- **Los contenedores de imágenes:**

- Deben tener un margen de 10px.
- Altura y ancho de 400px.
- El borde del contenedor debe ser de color gris y sólido.

- **Las imágenes:**

- Se deben cargar las imágenes desde la carpeta *images/*.

- Deben agregar un texto alternativo que sea igual al título de cada gráfico (¿Por qué agregar esto?).
- Todas las imágenes deben tener el mismo tamaño (Ojo que deben cumplir con este requisito usando reglas de CSS y no editando las imágenes con un editor de imágenes).

### Vizualizando Peliculas

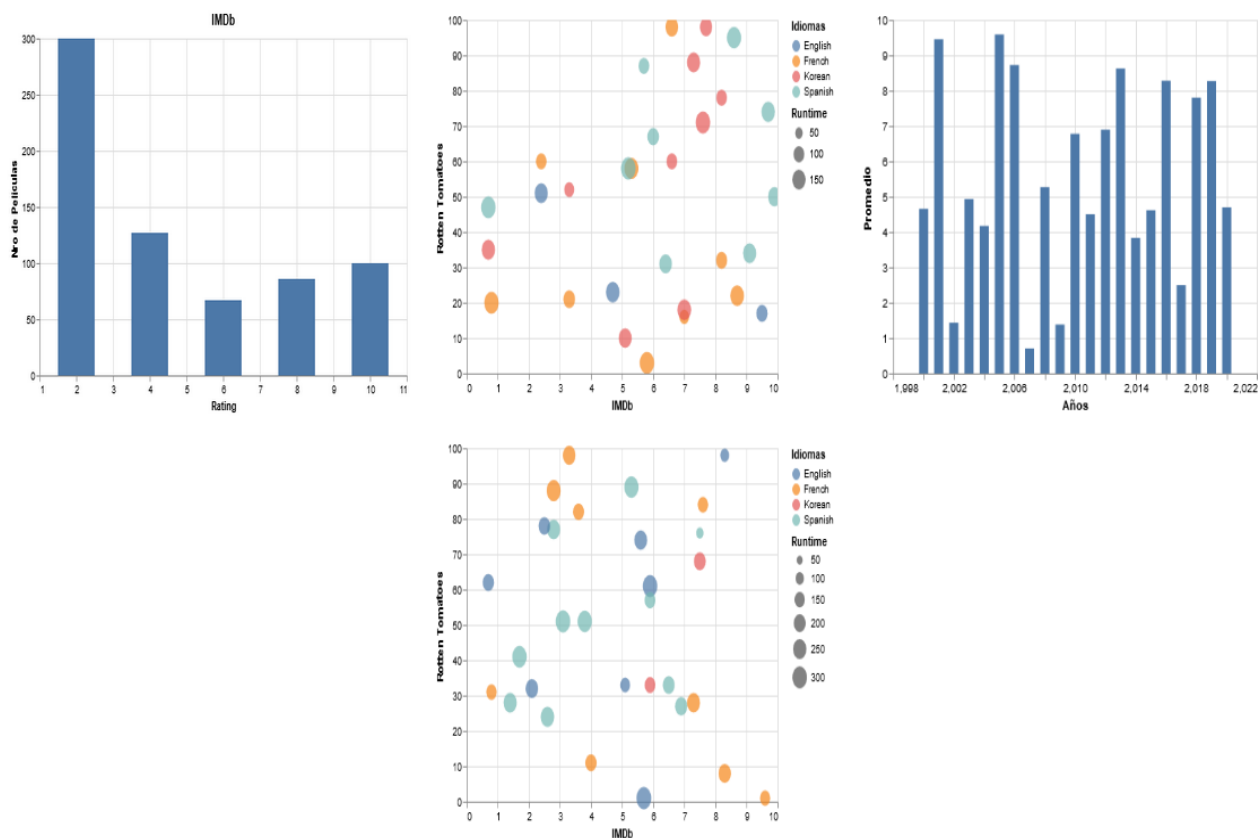


Figura 4: Página referencial de como se debería ver la visualización web.

### Parte 3: *Dark Mode e Interactividad* (2.5 puntos)

1. ¿Qué pasó con el botón 'DarkMode'? Hoy en día se ha vuelto popular tener la opción de convertir una página o *app* en modo oscuro. Bueno, esta sección esta destinada para que jueguen un poco con *Javascript* y le den ese comportamiento a su página. Al presionar en el botón '*DarkMode*', debe ocurrir lo siguiente:
  - El cuerpo debe cambiar de color a **negro**.
  - El encabezado debe cambiar de color a **gris** .
  - El botón debe cambiar su texto a '*NormalMode*'.

Después queremos poder volver a nuestro modo normal de la página (Sin recargarla, pruébenlo si quieren). Al presionar en el botón '*NormalMode*', debe ocurrir lo siguiente:

- El cuerpo debe cambiar de color a **#fafbfc**.
- El encabezado debe cambiar de color a **negro**.
- El botón debe cambiar su texto a '*DarkMode*'

**Nota:** Para poder asegurarnos de que no tengan ningún problema al agregar *JS* a la página, deberán abrir la consola (CMD) en **la carpeta donde esté el archivo HTML de la página** y correr el siguiente comando '*python -m http.server*', el cual les entregará el puerto en que están corriendo el servidor de python que acaban de crear.

Luego, en su navegador deben colocar la URL '*localhost:XXXX/YYYYY.html*', donde '*XXXX*' es el puerto del servidor y '*YYYYY*' es el nombre del archivo *HTML*.

2. Notarán que en la primera parte les pedimos hacer un total de 12 gráficos, sin embargo sólo pedimos que muestren 3. Esto se debe a que queremos que se muestre un gráfico de la pregunta 1 a la vez.

Se espera que usen *Javascript* para crear alguna especie de menú que nos permita cambiar qué gráfico de la primera pregunta se está mostrando de manera interactiva.



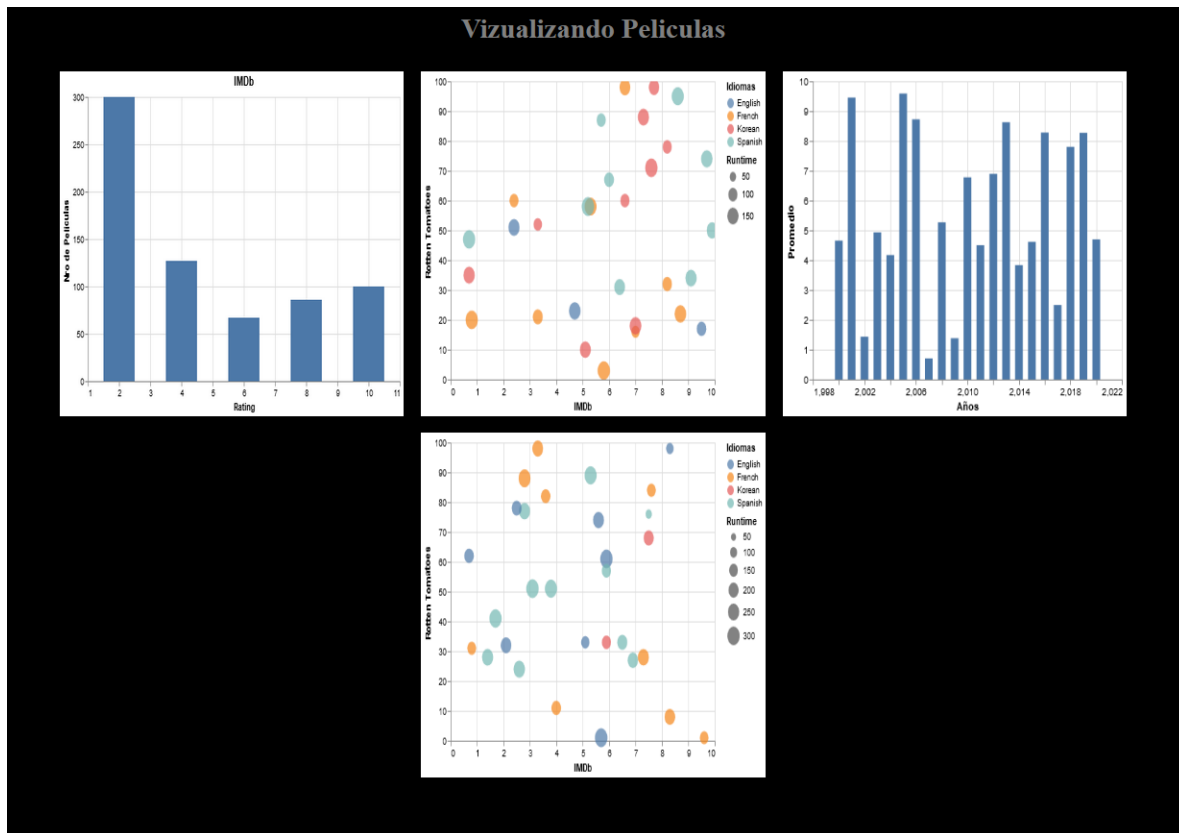


Figura 5: Página referencial de pagina en modo oscuro.

## Formato de entrega.

Para evaluar con facilidad la parte 1 de esta tarea, deberán realizarla en un *jupyter notebook* y subir este archivo de extensión `.ipynb` a su repositorio asignado.

En cuanto a la parte 2, deben incluir **todos** los archivos que permitan visualizar su página correctamente. En el repositorio base de la tarea, encontrarán una estructura propuesta que les ayudará a cumplir con el formato esperado.

## Entregas Atrasadas

Si así lo desea, existe la posibilidad de entregar la tarea de manera fuera de plazo, ya sea con 1 o 2 días de atraso.

Para esto, es necesario que envíen un mail a [reschilling@uc.cl](mailto:reschilling@uc.cl) a mas tardar a las 20:00 horas del día que deseen entregar. Las entregas serán recopiladas a las 23:59

y conllevarán un descuento de 2 puntos por día. No existirán excepciones a las reglas de atraso.

## Anexo

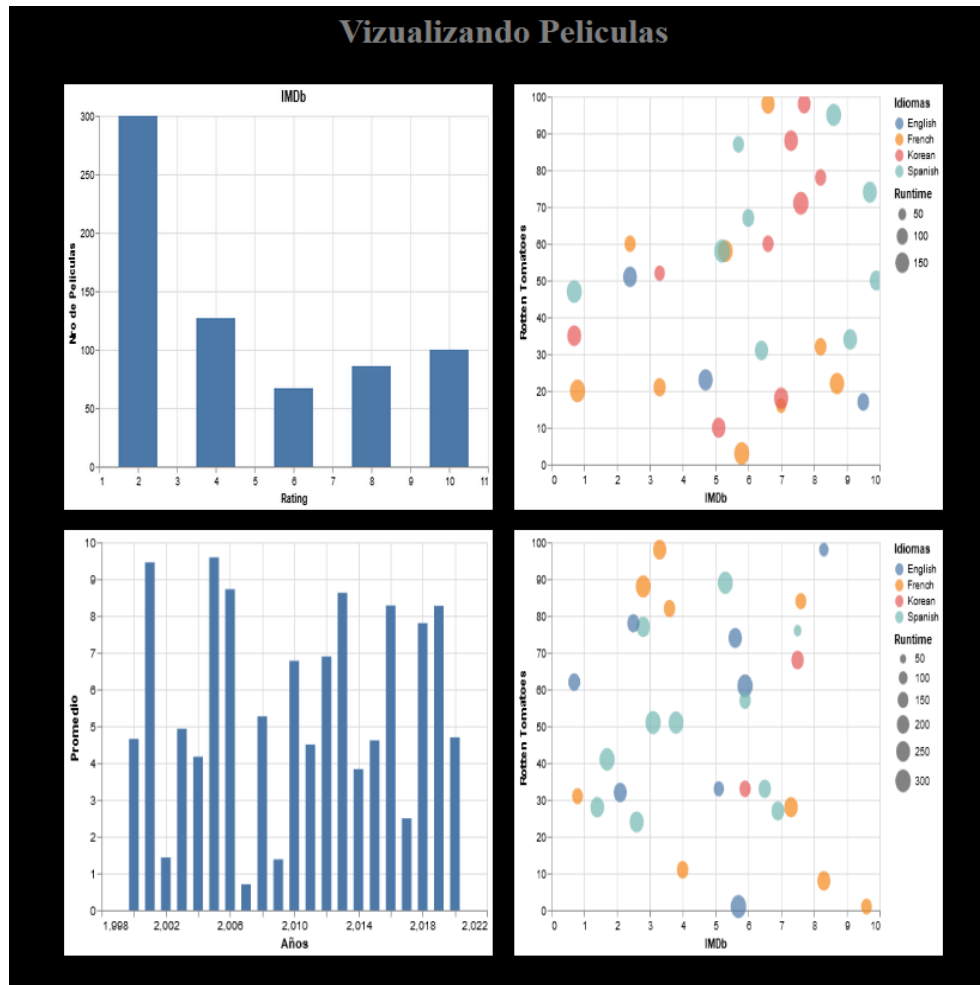


Figura 6: Página referencial con menos ancho, resulta en una grilla 2x2.

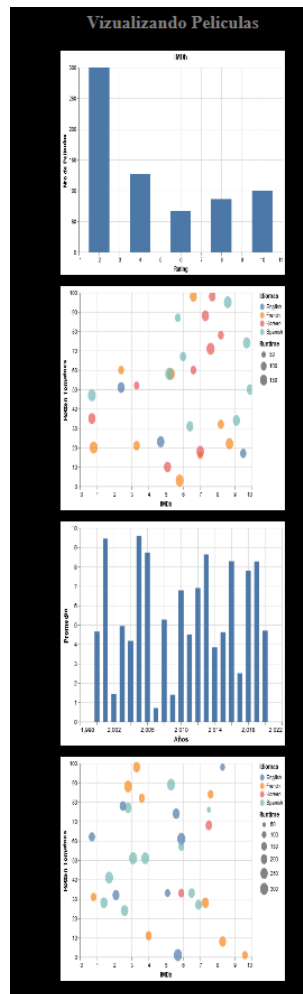


Figura 7: Página referencial con menos ancho, resulta en una grilla 4x1.