Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
Computación: Ciencia y Tecnología del Mundo Digital

Tarea Grande 1

Línea de comandos y Git
Ayudantes Luis Fros, Mauricio López y Javier Ruiz.
Profesor Denis Parra
Enunciado: 01 de abril de 2020

Indicaciones

- Fecha de Entrega: 14 de abril de 2020 hasta las 23:59
- Se debe entregar la tarea en el repositorio asignado a cada uno por Github Classroom.
- Cada hora (o fracción) de atraso descuenta 1.5 puntos de la nota que se obtenga.
- La tarea es *individual*. 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 aprendas a:

- Utilizar Python para manipular datos.
- Utilizar los datos para poder hacer gráficos con la librería Altair.
- Aprender las 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 tu tarea trabajarás con los datos entregados para obtener lo necesario para la confección de un dashboard. Es importante considerar:

- **El entregable de la sección 1 es un jupyter notebook (extensión.ipynb)**, donde se aprecie el **código, gráficos realizados y explicaciones** que te 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, tu nota máxima será un 4.
- Se espera que uses la librería Pandas de manera **eficiente** (por ejemplo, **evitando loops innecesarios**). De lo contrario, podrían haber descuentos en tu nota. Lo anterior es a criterio del ayudante.
- Los **gráficos deben realizarse obligatoriamente con la librería Altair**. Si no utilizas la librería anteriormente mencionada para las visualizaciones, **no obtendrás puntaje en los ítems relacionados con este punto.**
- Los gráficos deben tener título y nombres para cada eje según corresponda, si no se aplicará un descuento a cada gráfico que tenga esta falta.

Datos entregados

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

- **budget**: El presupuesto para la película.
- **company**: La compañía que produjo la película.
- **country**: País de Origen.
- **director**: El director de la película.
- **genre**: El género de la película
- **revenue**: Los ingresos monetarios generados por la película.

- **name**: El nombre de la película.
- **rating**: El rating de la película.
- **released**: Fecha en la que película fue lanzada. Está en formato (YYYY-MM-DD).
- **runtime**: Duración de la película.
- **score**: Puntaje/nota promedio de la película.
- **star**: Actor o actriz principal de la película.
- **votos**: El director de la película.
- **writer**: El escritor de la película.
- **year**: El año en el que se lanzó la película.

Instrucciones

Parte 1: manipulación y visualización de datos

Pregunta 1

Queremos analizar la distribución típica de los puntajes promedio de las películas (como para ver, por ejemplo, qué nota tienden a poner los usuarios). Para esto, realiza un **gráfico de barras** que en su eje x tenga las notas desde 1 a 10, y que en el eje y tenga la cantidad de películas que tienen esa nota.

Para realizar este gráfico, debes 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).

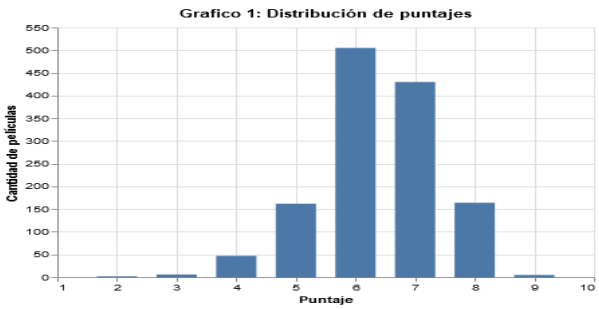

Figura 1: Ejemplo del gráfico de barras

Pregunta 2

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

Realiza un gráfico de barras que esté compuesto por las mejores **10** películas según su puntaje. En su eje *x*, deben estar los nombres de cada una de las 10 películas, y en el eje *y* debe estar el puntaje de cada una de ellas. Queremos ver rápidamente cuál veremos primero y por cuáles seguiremos. Es por esto que las películas deben estar de izquierda a derecha, de mayor a menor puntaje, y la barra de la película con el mejor puntaje debe estar pintada de un color distinto el resto.

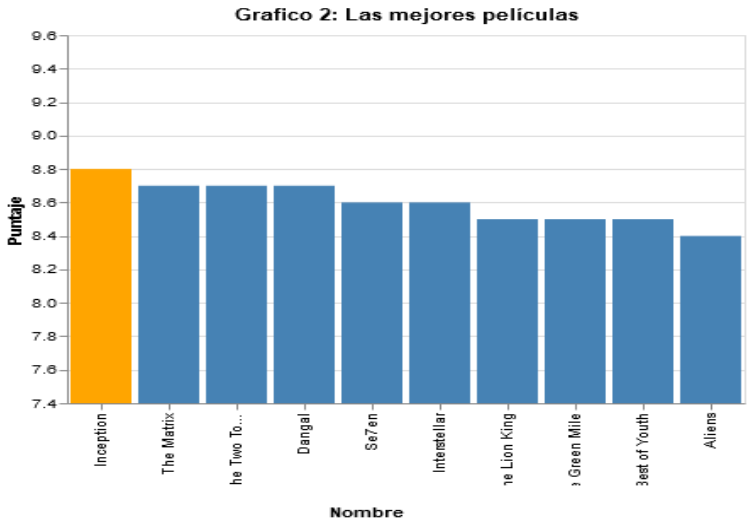

Figura 2: Ejemplo del gráfico de barras

Pregunta 3

Algo interesante para ver es si el dinero utilizado en realizar la película influye efectivamente en la nota final. Para eso, realiza un diagrama de dispersión que en el eje *y* tenga el puntaje y en el eje *x* tenga el presupuesto (budget).

Para realizar el gráfico de manera correcta, no deben aparecer las películas que marcan 0 en su presupuesto.

¿Dirías que el presupuesto es determinante del puntaje?

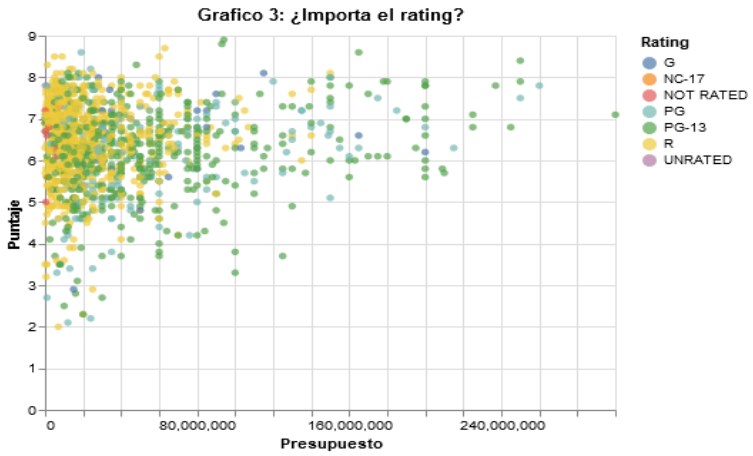

Figura 3: Ejemplo del diagrama de dispersión

Pregunta 4

También sería interesante saber si el promedio del puntaje de las películas se relaciona con el mes del año con el que fue lanzado. Realiza un gráfico de barras en el que los meses se encuentren en el eje x y que el promedio de las películas lanzadas durante mes se encuentren en el eje y.

Para realizar este gráfico, no debes considerar las películas que no tengan mes de lanzamiento.

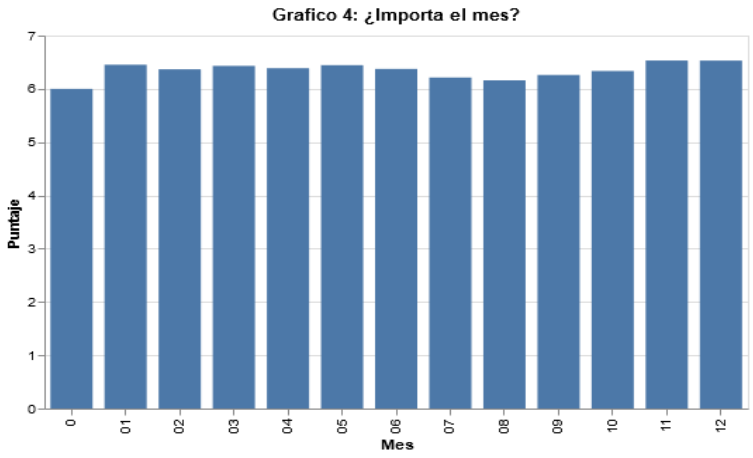

Figura 4: Ejemplo del gráfico de barras

Parte 2: Comunicación con API de wikipedia

En esta parte, tu código de Python deberá comunicarse con la interfaz que provee Wikipedia para encontrar más información acerca de algunas películas. Se puede obtener, por ejemplo, información resumida de la película. Para lo anterior, ocuparemos la librería Wikipedia de python, lo que nos permitirá conectarnos fácilmente. Obtén el resumen de wikipedia (el *summary*) de cada una de las 5 mejores películas, utilizando la librería anteriormente descrita. Debes almacenar esta información dentro de un dataframe, junto con el nombre de la película. Este dataframe puede tener más columnas y eso no llevará descuento.

Parte 3: *dashboard* (2.5 puntos)

Con las visualizaciones ya generadas, debes construir una página web que contenga cada uno de los gráficos mencionados en el enunciado. Se recomienda revisar la siguiente guia de como funciona *Flexbox* Como la confección de los gráficos será evaluada en la parte 1, en esta parte se espera que cumplas con lo siguiente:

1. Lista de gráficos: La pagina debe mostrar una lista de cuatro gráficos. Los cuales deben estar centrados y alineados. A continuación se detallan los requisitos de cada elemento HTML en la pagina:
   - 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**
   - El botón
     - Debe tener los bordes redondos con 7px.
   - El contenedor que contiene a todas las imagenes
     - Debe ser un contenedor *flex*
     - Los elementos dentro deben estar **centrados**
     - Los elementos dentro deben saltar de linea 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 imagenes:
  - Deben tener un margen de 10px.
  - Altura y ancho de 400px
  - El borde del contenedor debe ser de color gris y solido.
- Las imagenes:
  - Se deben cargar las imágenes desde la carpeta `images/`
  - Deben agregar un texto alternativo que sea igual al titulo de cada gráfico. (¿Porque 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).

Figura 5: Pagina referencial de como se debería ver la visualización web.

¡Bonus! (0,5 puntos)

Si la tarea te resultó especialmente motivadora y quieres seguir aprendiendo, te proponemos los siguientes desafíos, que permitirán que puedas obtener como **nota máxima un 7.5**:

1. ¿Que paso con el botón 'DarkMode' ? Hoy en día se ha vuelto popular tener la opción de convertir una pagina o app en modo oscuro. Bueno, esta sección del bonus esta destinada a que juegues un poco con Javascript y le des ese comportamiento a nuestra pagina. Al presionar en el botón 'DarkMode', deben 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 pagina (Sin recargar la pagina, pruébalo si quieres). Al presionar en el botón 'NormalMode', deben 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 que no tengas ningún problema al agregar JS a la página es que deberás 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 te entregará el puerto en que está corriendo el servidor de python que acabamos de crear.
   Luego, en tu navegador debes colocar la URL *'localhost:XXXX/YYYYY.html'*, donde 'XXXX' es el puerto del servidor y 'YYYYY' es el nombre del archivo HTML.

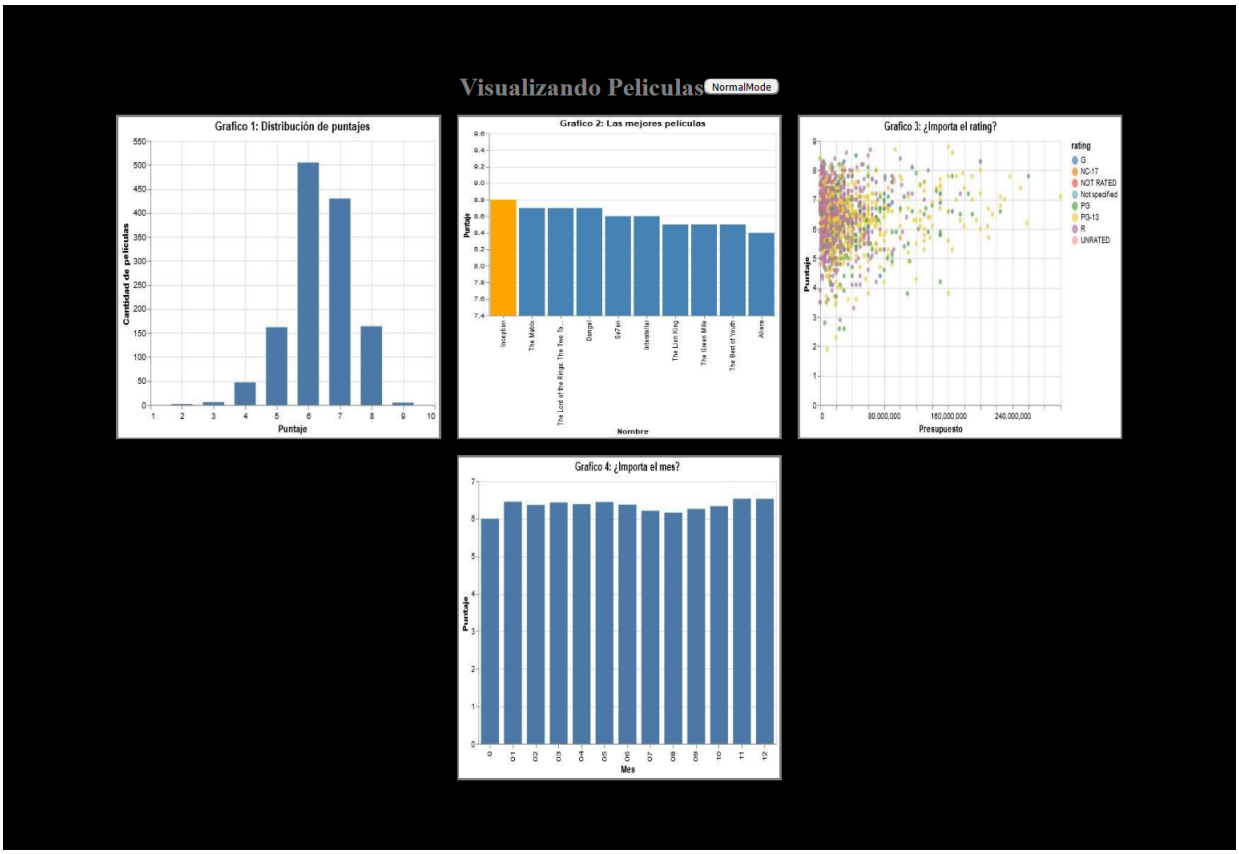

Figura 6: Pagina referencial de pagina en modo oscuro.

Formato de entrega

Para evaluar con facilidad la parte 1 de esta tarea, deberás realizarla en un *jupyter notebook* y subir este archivo de extensión `.ipynb` a tu repositorio asignado. En cuanto a la parte 2, debes incluir **todos** los archivos que permitan visualizar tu página correctamente. En el repositorio base de la tarea, encontrarás una estructura propuesta que te ayudará a cumplir con el formato esperado.

Anexo

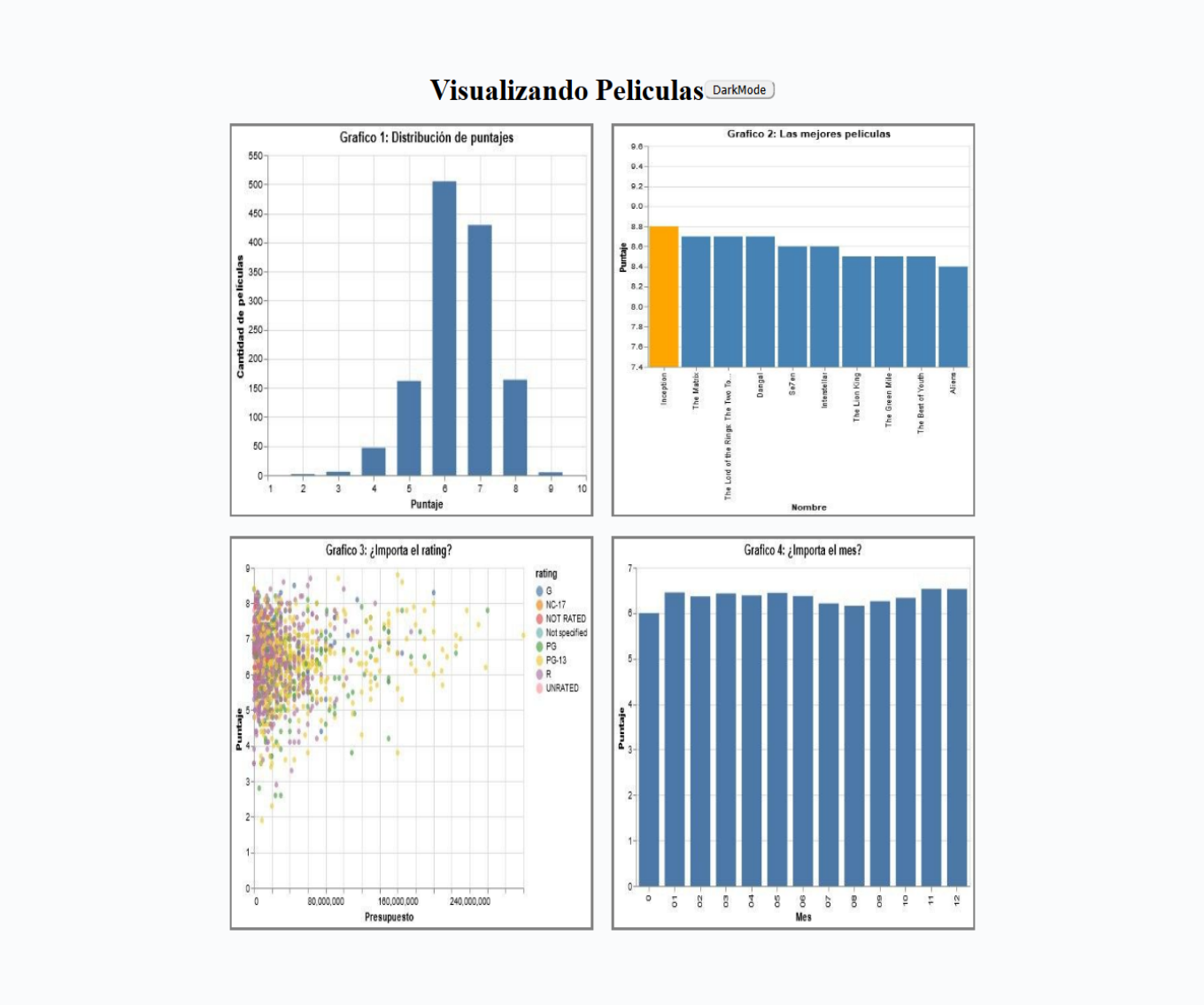

Figura 7: Pagina referencial con menos ancho, resulta en una grilla 2x2.

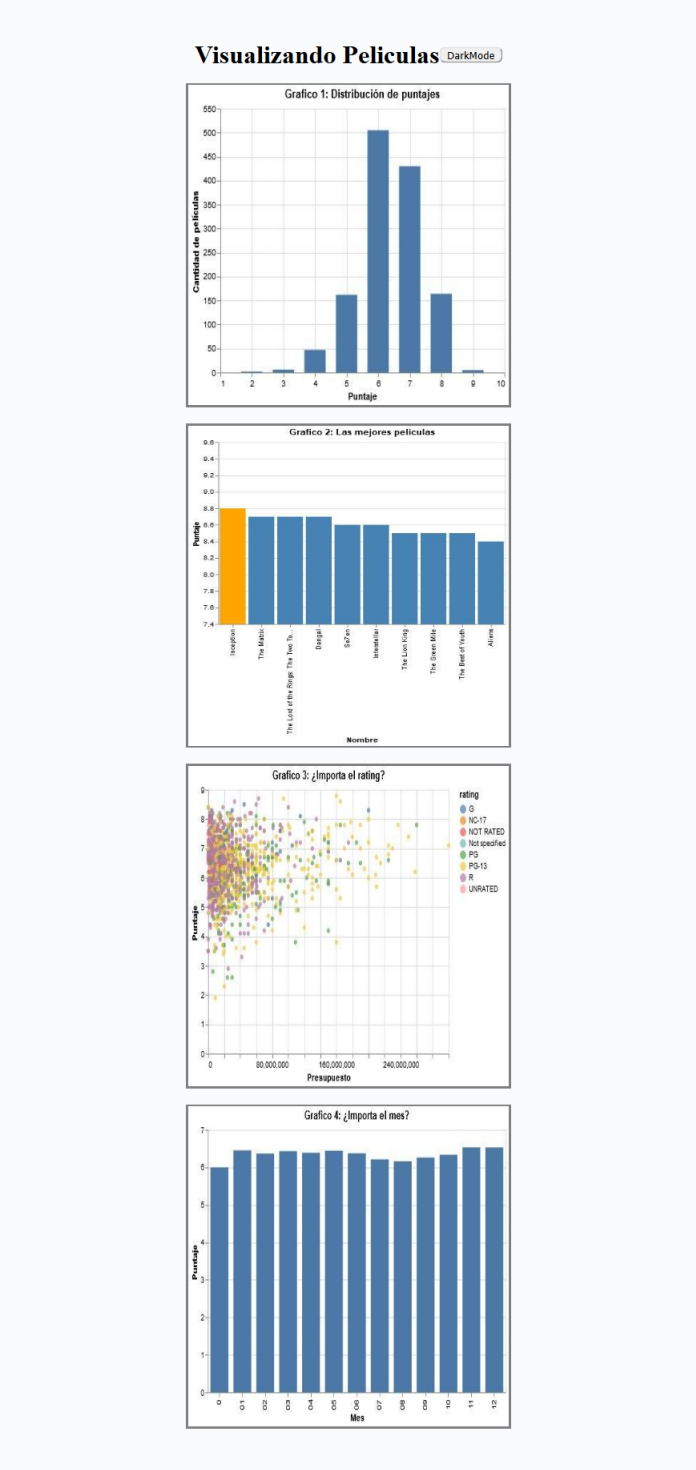

Figura 8: Pagina referencial con menos ancho, resulta en una grilla 4x1.