

# Tarea Grande 1

Profesores Vicente Domínguez y Luis Ramírez

Anunciada: 19 de agosto de 2019

---

## Indicaciones

- Fecha de Entrega: 02 de septiembre de 2019.
  - Debes entregar tu tarea en tu repositorio GitHub privado asignado para esta evaluación.
  - Cada hora o fracción de atraso descuenta 0,5 puntos de la nota que obtengas.
  - La tarea es *individual*. La copia será evaluada con nota 1 en el la tarea, además de las sanciones disciplinarias correspondientes.
- 

## Objetivo

El objetivo de esta tarea es que aprendas a:

- Utilizar Python para manipular datos.
- Crear una página web sencilla que contemple el uso de HTML, CSS y Javascript, para confeccionar un *dashboard*.
- Utilizar de manera básica una API de un servicio externo.

En particular, tendrás que manipular datos de los hábitos musicales de un usuario real de [LastFM](#), para utilizarlos en la confección de un *dashboard* que muestre información interesante sobre la persona estudiada.

## Descripción de los datos

Junto a este enunciado, se te entregaron dos archivos más:

1. `dataset_final.csv`: contiene todas las canciones que fueron escuchadas por un usuario desde enero de 2017 hasta el 21 de marzo de 2018. Los campos que contiene el archivo son:
  - a) Date: fecha en que se escuchó la canción, en formato *Unix Epoch Time*.
  - b) Track: título de la canción.
  - c) Artist: intérprete de la canción.
2. `artists_genres_images.csv`: archivo que contiene el género musical y la imagen de portada de todos los artistas presentes en el archivo anterior. Sus campos son:
  - a) Artist: artista.
  - b) Genre: género musical.
  - c) Image\_Link: enlace a la imagen de portada del artista.

## Instrucciones

### Parte 1: Manipulación de los datos y generación de gráficos con Python (3.8 puntos)

En la primera parte de tu tarea, trabajarás con los datos entregados para obtener lo necesario para la confección del *dashboard*. **El entregable de esta sección es un *jupyter notebook* (extensión `.ipynb`)**. De no cumplir con el tipo de archivo, recibirás una penalización en tu nota. Además, **debes ocupar** las librerías `pandas` y `matplotlib` para realizar lo siguiente:

1. Convertir las fechas a un formato más legible (YYYY-mm-DD HH:MM), como por ejemplo 2018-03-21 23:03:17 (**0.2 puntos**). Asegúrate de que las horas estén en el huso horario GMT -4 (hora chilena).
2. Confeccionar un gráfico de barras donde en el eje  $x$  se vean los nombres de las 10 canciones más escuchadas y en el  $y$  la cantidad de veces que fue escuchada cada una (**0.5 puntos**).
3. Obtener los 5 artistas más escuchados y disponer esta información en un DataFrame cuyas columnas sean: nombre del artista, veces que fue escuchado y enlace a su imagen<sup>1</sup>. Posteriormente, guardar esta información en un archivo `.csv`, generado con Python (**0.3 puntos**).
4. Generar un gráfico de torta de los géneros musicales más escuchados por el usuario. Deben agregar los nombres y valores porcentuales a cada género en el gráfico. (**0.5 puntos**).

---

<sup>1</sup>Para este ítem, es conveniente que investigues sobre el método `merge` de `pandas`.

5. Graficar la evolución de la cantidad de canciones escuchadas por mes (**0.7 puntos**). [Ejemplo](#).
6. Realizar un gráfico de torta donde se evidencie la cantidad de canciones escuchadas según día de la semana<sup>2</sup>. Se debe poder identificar cuántas canciones fueron escuchadas y en qué día se realizó esa acción. (**0.8 puntos**).
7. Generar un gráfico de barras en el que se vea cuántas canciones se escucharon según hora del día<sup>3</sup> (**0.8 puntos**). [Ejemplo](#).

**En caso de no cumplir con el uso de las librerías mencionadas anteriormente para obtener y graficar la información pedida, tendrás un descuento de un (1) punto en tu nota de la tarea.**

**Nota:** para poder utilizar los gráficos en la Parte 3, estos deben ser exportados a un archivo de imagen.

## Parte 2: Comunicación con API de Spotify (0.5 puntos)

En esta parte, tu código de Python deberá comunicarse con la interfaz que provee Spotify para encontrar más información acerca de las canciones entregadas. Se puede obtener información muy interesante de las canciones, tales como *tempo*, *bailabilidad*<sup>4</sup>, y la instrumentalidad. Para lo anterior utilizaremos [Spotipy](#), una librería de Python que facilitará la utilización de las funciones de la API de Spotify. Tendrás el puntaje completo en esta parte si utilizas la librería mencionada para obtener la *bailabilidad* (*danzability*) de las 5 canciones más escuchadas del *dataset*. Esta información debe ser almacenada en un DataFrame de *pandas* que debe ser impreso en el mismo *jupyter notebook* de la parte 1 (**0.5 puntos**).

**Nota:** como seguramente sabes, la Universidad tiene bloqueado el acceso a Spotify, por lo que esta parte no podrá ser desarrollada utilizando la red Eduroam dentro de los campus de la misma. Nuestra recomendación es trabajar desde fuera de la Universidad o, si estás en San Joaquín, acudir al Work Café del Banco Santander o al Starbucks del campus y utilizar las redes Wifi que ellos proveen. Compartir internet desde tu celular a tu computador también es una opción para trabajar dentro de la Universidad.

## Parte 3: Dashboard (1.7 puntos)

Con los datos y gráficos ya obtenidos, debes construir una página web que contenga lo siguiente:

---

<sup>2</sup>Lunes, martes, miércoles, jueves y viernes

<sup>3</sup>Para este ítem y los dos anteriores, se recomienda emplear los métodos `set_index` y `resample` de *pandas*.

<sup>4</sup>Sí, esto es una métrica de cuán *bailable* es una canción

1. Gráfico de las 10 canciones más escuchadas.
2. Gráfico de los géneros musicales más escuchados.
3. Gráfico de la evolución mensual de la cantidad de canciones escuchadas.
4. Gráfico de la cantidad de canciones escuchadas según día de la semana.
5. Gráfico de la cantidad de canciones escuchadas según hora del día.

Para las siguientes instrucciones, **no** se permite la utilización de librerías o *frameworks* externos. Se espera que se ajuste la posición de las imágenes **sólo** utilizando reglas de CSS. Como la confección de los gráficos será evaluada en la parte 1, para esta parte se evaluará lo siguiente:

1. La página debe ser *responsive*, los elementos que la componen están correctamente alineados y no se superponen unos con otros (**0.5 puntos**).  
Para probar que tu página es *responsive*, puedes probar reduciendo el tamaño de la ventana del navegador. Las imágenes deberían ajustarse al tamaño de la pantalla, desplazándose cuando corresponda para no tener que hacer *scroll* horizontal, sino solo de manera vertical, en caso de no caber más de una imagen en la misma línea.
2. La página, estando maximizada, sigue la siguiente disposición de las imágenes de los gráficos (**0.7 puntos**):



Mostrar/ocultar

Gráfico 1: Ejemplo



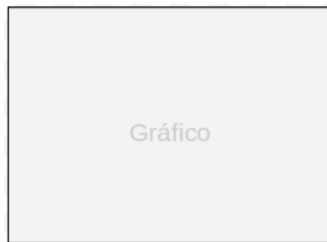
Mostrar/ocultar

Gráfico 2: Ejemplo



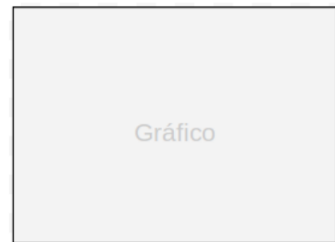
Mostrar/ocultar

Gráfico 3: Ejemplo



Mostrar/ocultar

Gráfico 4: Ejemplo



Mostrar/ocultar

Gráfico 5: Ejemplo

Para esta parte, **no** se permite la utilización de librerías o *frameworks* externos. Se espera que se ajuste la posición de las imágenes **exclusivamente** utilizando reglas de CSS. No cumplir con esta regla, generará un descuento de 0.5 puntos de la nota final de tu tarea.

3. Botón mostrar/ocultar bajo cada gráfico, que al ser presionado, muestre una breve descripción del gráfico, de la siguiente forma (**0.5 puntos**):



Mostrar/ocultar



Mostrar/ocultar

Gráfico 1: Ejemplo

## Entregables

Para la evaluación de la tarea, deberás subir a tu repositorio asignado:

1. Archivo de *Jupyter Notebook*, con extensión `.ipynb`, que contenga el desarrollo de la parte 1 y 2.
2. Archivo HTML, junto con todos los archivos que sean necesarios para poder visualizar la página correctamente (que al abrir el archivo HTML, se puedan ver todos los gráficos y lo solicitado en la parte 3).