

## N2 – PROY: CupiTikTok

**IMPORTANTE:** Este proyecto debe realizarse de forma **completamente individual**.

### Objetivo general

Practicar los conceptos clave estudiados en el Nivel 2 del curso.

### Objetivos específicos

1. Llamar a funciones con parámetros y componer funciones.
2. Usar la técnica *Dividir y Conquistar*.
3. Crear y usar un módulo.
4. Practicar condicionales.
5. Utilizar diccionarios.
6. Construir interfaces de usuario basadas en consola.

### Actividad 1 | Entendimiento del problema

#### CupiTikTok

CupiTikTok es una red social reconocida por sus retos y bailes virales. Un grupo de *creadores de contenido* (usuarios de la red que producen contenido y lo publican en la misma) le ha pedido crear un programa que les permita hacer un análisis de sus perfiles y posición en la red; para esto se requiere que la aplicación sea capaz de:

- Buscar un creador de contenido por su nombre.
- Filtrar a los creadores de contenido que pertenecen a una categoría específica.
- Calcular el promedio de vistas de los creadores de contenido.
- Filtrar a los creadores de contenido que superen un mínimo de vistas (umbral).
- Calcular el *rating* de un creador de contenido.
- Buscar al creador que más se ajusta a los gustos del usuario según un puntaje de afinidad.
- Buscar al creador de contenido que más días lleva sin publicar.

## Cálculo del *rating* de un creador de contenido

El *rating* de un creador de contenido se calcula usando la fórmula *F1*:

$$\text{rating} = \left( \left( \frac{\text{número de seguidores}}{S_{\text{máx}}} \right) \cdot 0.5 \right) + \left( \left( \frac{\text{número de likes}}{L_{\text{máx}}} \right) \cdot 0.3 \right) + \left( \left( \frac{\text{número de vistas}}{V_{\text{máx}}} \right) \cdot 0.2 \right)$$

$$\begin{aligned} S_{\text{máx}} &= 600\,000 \\ L_{\text{máx}} &= 100\,000\,000 \\ V_{\text{máx}} &= 100\,000\,000 \end{aligned}$$

*F1* Calcula un *rating* que permite saber qué tan relevante es un creador de contenido en la red social con base en su número de seguidores, *likes* (reacciones de “me gusta”) y vistas.

Las constantes  $S_{\text{máx}}$ ,  $L_{\text{máx}}$  y  $V_{\text{máx}}$  son una simplificación. Los algoritmos usados por redes sociales reales son muy complejos y suelen estar protegidos por figuras de derechos de autor. Los resultados de esta fórmula siempre son positivos y suelen estar entre 0 y 100.

## Cálculo del puntaje de afinidad

El puntaje de afinidad entre un creador de contenido y un usuario se calcula en función de tres aspectos: las categorías de contenido que el creador produce, su país de origen y su *rating*. Entonces, el *rating* se refiere a la relevancia de un creador de contenido en la red social, mientras que el puntaje de afinidad describe cuantitativamente la afinidad entre un creador y un usuario. A continuación, se explica cómo se calcula el puntaje de afinidad:

- Si el creador hace contenido de la categoría de interés del usuario se le suman **3** puntos, en caso contrario no pasa nada.
- Si el creador de contenido es del país de interés del usuario se le suman **2** puntos, en caso contrario se le resta **1**.
- Si el creador de contenido tiene un *rating* inferior a lo que busca el usuario, se le restan **5** puntos, en caso contrario se le suman **2**.

Cada creador de contenido será representado por un diccionario cuyas llaves son cadenas de caracteres en minúsculas y sin tildes. Los valores pueden incluir tildes y variar en mayúsculas o minúsculas. A continuación, se presenta una tabla explicativa de cada llave y su descripción. Es muy importante que escriba las llaves exactamente como aparecen en la tabla.

Llave	Tipo	Descripción de la llave
<b>nombre</b>	str	Nombre del creador de contenido. Por ejemplo: "Code_Destroyer2002" sin espacios.
<b>país</b>	str	País de origen del creador de contenido.
<b>categorías</b>	str	Categorías a las que pertenece el contenido separado por punto y coma (;) sin espacios. Los valores para las categorías posibles son: - Comedia - Educación - Baile - Maquillaje - Cocina - Lectura
<b>likes</b>	int	Número total de <i>likes</i> acumulados por el creador de contenido.
<b>vistas</b>	int	Número total de vistas acumuladas del creador de contenido.
<b>seguidores</b>	int	Número de seguidores que tiene el creador de contenido.
<b>fecha_ultima_publicacion</b>	int	Fecha de la última publicación hecha por el creador de contenido, representada como un entero en el formato: YYYYMMDD. Por ejemplo: 20240223.

## Actividad 2 | Preparación del ambiente de trabajo

1. Cree una carpeta para trabajar, poniéndole como nombre su *login* Uniandes.
2. Descargue de Bloque Neón el archivo con el esqueleto del proyecto (n2-esqueleto.zip). El esqueleto consiste en un conjunto de archivos que usted va a usar y a modificar.
3. Descomprima el esqueleto en la carpeta que usted acaba de crear.
4. Abra Spyder y cambie la **carpeta de trabajo** (*working directory*) para que sea la carpeta con el esqueleto.

## Actividad 3 | Construir el módulo de lógica

5. Usando Spyder, abra y complete el archivo `cupi_tiktok.py`. Este archivo tendrá el módulo con la lógica del programa, es decir, las funciones que se encargarán de realizar los cálculos descritos. Note que las funciones de este módulo están debidamente documentadas y tienen etiquetas TODO (por hacer) numeradas (1-8), indicando que usted debe completarlas. Por favor reemplace la instrucción `pass` con su implementación de la función según la documentación.

A continuación, se provee una descripción detallada que guiará su implementación de la lógica. **Es fundamental evitar la duplicación de código. Determine si una función debe llamar a otra (composición) para evitar que sus funciones contengan código repetido.**

<b>Nombre de la función</b>		buscar_creador_por_nombre
<b>Descripción de la función</b>		Busca si entre los creadores de contenido dados, hay uno con el nombre ingresado.
<b>Parámetros</b>		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
nombre	str	Nombre del creador de contenido a buscar.
c1	dict	Diccionario que representa al primer creador de contenido.
c2	dict	Diccionario que representa al segundo creador de contenido.
c3	dict	Diccionario que representa al tercer creador de contenido.
c4	dict	Diccionario que representa al cuarto creador de contenido.
Retorno	dict	Diccionario del creador de contenido con el nombre ingresado por el usuario. Retorna <b>None</b> si no hay ningún creador con ese nombre. Si dos creadores o más tienen el mismo nombre, retorna el primero encontrado.

<b>Nombre de la función</b>		filtrar_creadores_por_categoria
<b>Descripción de la función</b>		Filtra a los creadores de contenido dados según la categoría ingresada.
<b>Parámetros</b>		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
categoría	str	Categoría de interés para el usuario.
c1	dict	Diccionario que representa al primer creador de contenido.
c2	dict	Diccionario que representa al segundo creador de contenido.
c3	dict	Diccionario que representa al tercer creador de contenido.
c4	dict	Diccionario que representa al cuarto creador de contenido.
Retorno	str	Nombres de los creadores que hacen contenido de la categoría dada, separados por comas ( , ). Retorna <b>None</b> si no hay ningún creador que haga parte de la categoría de interés ingresada.

<b>Nombre de la función</b>		<b>calcular_promedio_vistas</b>
<b>Descripción de la función</b>		Calcula el promedio de vistas de todos los creadores de contenido dados.
<b>Parámetros</b>		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
c1	dict	Diccionario que representa al primer creador de contenido.
c2	dict	Diccionario que representa al segundo creador de contenido.
c3	dict	Diccionario que representa al tercer creador de contenido.
c4	dict	Diccionario que representa al cuarto creador de contenido.
Retorno	float	Promedio de vistas de todos los creadores de contenido dados, redondeado a dos cifras decimales.

<b>Nombre de la función</b>		<b>filtrar_creadores_por_vistas</b>
<b>Descripción de la función</b>		Filtra a los creadores de contenido dados que superen un mínimo de vistas (umbral).
<b>Parámetros</b>		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
minimo_vistas	int	Umbral de vistas.
c1	dict	Diccionario que representa al primer creador de contenido.
c2	dict	Diccionario que representa al segundo creador de contenido.
c3	dict	Diccionario que representa al tercer creador de contenido.
c4	dict	Diccionario que representa al cuarto creador de contenido.
Retorno	str	Nombres de los creadores de contenido dados que superan el mínimo de vistas ingresado, separados por comas (,). Retorna el mensaje "Ninguno" si ningún creador de contenido supera el umbral.

<b>Nombre de la función</b>		calcular_rating_creador
<b>Descripción de la función</b>		Calcula el <i>rating</i> de un creador de contenido con base en su número de seguidores, <i>likes</i> y vistas usando la fórmula <i>F1</i> .
<b>Parámetros</b>		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
creador	dict	Diccionario que representa a un creador de contenido.
Retorno	float	El <i>rating</i> del creador de contenido, redondeado a dos cifras decimales. Este valor se encuentra entre 0 y 100.

<b>Nombre de la función</b>		calcular_puntaje_afinidad
<b>Descripción de la función</b>		Calcula el puntaje de afinidad entre un creador de contenido y un usuario basándose en una categoría, un país y un <i>rating</i> mínimo dados (ver sección: <i>Cálculo del puntaje de afinidad</i> , en la página 2).
<b>Parámetros</b>		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
creador	dict	Diccionario que representa a un creador de contenido.
categoria	str	Categoría de interés para el usuario.
minimo_rating	float	<i>Rating</i> mínimo que el creador de contenido debe superar.
pais	str	Nombre del país de interés para el usuario.
Retorno	float	El puntaje de afinidad que tiene un creador con un usuario, redondeado a dos cifras decimales.

<b>Nombre de la función</b>		buscar_creador_favorito
<b>Descripción de la función</b>		Busca el creador de contenido con el mayor puntaje de afinidad, teniendo en cuenta la categoría, el país y un <i>rating</i> mínimo especificados.
<b>Parámetros</b>		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
categoria	str	Categoría de interés para el usuario.
rating	float	Rating mínimo que debe tener el creador de contenido.
país	str	Nombre del país de interés para el usuario.
c1	dict	Diccionario que representa al primer creador de contenido.
c2	dict	Diccionario que representa al segundo creador de contenido.
c3	dict	Diccionario que representa al tercer creador de contenido.
c4	dict	Diccionario que representa al cuarto creador de contenido.
Retorno	str	<p>El nombre del creador de contenido con mayor puntaje de afinidad y su puntaje obtenido como sigue:  "{nombre} con puntaje {puntaje_afinidad}", por ejemplo:  "Code_Destroyer2002 con puntaje 100".</p> <p>Si dos creadores resultan con el mismo puntaje de afinidad, se retorna la información del que tenga el nombre alfabéticamente anterior/menor (considerando el orden de sus caracteres en el abecedario).</p>

<b>Nombre de la función</b>		buscar_creador_inactivo
<b>Descripción de la función</b>		<p>Busca al creador de contenido que lleva el mayor tiempo sin publicar con base en una fecha ingresada por parámetro y la llave <code>fecha_ultima_publicacion</code>.</p> <p><b>Nota:</b> Para facilidad de los cálculos debe suponer que cada mes tiene 30 días y que el año tiene 360 días.</p>
<b>Parámetros</b>		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
<code>fecha_de_referencia</code>	<code>int</code>	Fecha con la que se busca hacer el análisis de actividad en formato YYYYMMDD.
<code>c1</code>	<code>dict</code>	Diccionario que representa al primer creador de contenido.
<code>c2</code>	<code>dict</code>	Diccionario que representa al segundo creador de contenido.
<code>c3</code>	<code>dict</code>	Diccionario que representa al tercer creador de contenido.
<code>c4</code>	<code>dict</code>	Diccionario que representa al cuarto creador de contenido.
Retorno	<code>dict</code>	Diccionario con las llaves: "nombre", "años", "meses" y "días", cuyos valores son respectivamente el nombre del creador más inactivo y el tiempo de inactividad en años, meses y días. Si dos creadores tienen el mismo tiempo sin publicar, retorna el que tenga menos seguidores. Si la fecha ingresada es anterior a todas las fechas de última publicación, retorna <code>None</code> .



## Actividad 4 | Construir interfaz de usuario basada en consola

6. En esta actividad, se le solicita construir la interfaz de usuario basada en consola para que el usuario interactúe con su programa completando el archivo `consola_cupi_tiktok.py` que se encuentra en la carpeta `n2-esqueleto`. Note que las funciones de este módulo están debidamente documentadas y tienen etiquetas `TODO`, numeradas (9-15) indicando que usted debe completarlas. Por favor reemplace la instrucción `pass` con su implementación de la función según la documentación.

**ATENCIÓN:** Su interfaz debe seguir el estándar de construcción de consolas visto en clase.

**ATENCIÓN:** Su interfaz debe solicitar al usuario todos los valores de argumentos requeridos para invocar correctamente a las funciones de la lógica.

7. Si una función presenta errores o resultados inesperados, revise su implementación. Los errores comunes incluyen nombres incorrectos, argumentos insuficientes, en exceso, o en el orden equivocado. Repita las pruebas tras cada corrección.

**ATENCIÓN:** Cuantas más pruebas (de casos normales, extremos y anormales) realice, más indicios tendrá de que su programa está bien construido.

## Entrega

8. Comprima la carpeta con su proyecto resuelto. El archivo debe llamarse **N2-PROY-*login*.zip**, donde *login* es su nombre de usuario de Uniandes (Por ejemplo: `N2-PROY-p.perez123.zip`).

**ATENCIÓN:** El archivo debe ser un comprimido del tipo `.zip`.

9. Entregue el archivo comprimido a través de Bloque Neón en la tarea designada como **Proyecto del Nivel 2**.

**IMPORTANTE:** Recuerde que su entrega con la solución del proyecto debe ser de su completa autoría.