Proyecto de clase de **Python aplicado**.

Fundamentos de Programación

**Nombre del Proyecto**

**1. Información General**

* **Nombre estudiantes:**

**Daniel Garzon**

**Nicolas Fayad**

**Camilo Saenz**

**Javier Russi**

* **Curso / Grupo:A**
* **Fecha de entrega:16 OCT 2025**
* **Profesor:Pablo Carreño**

**2. Título del Proyecto**

CineMatch

**3. Descripción del Proyecto**

Breve resumen del proyecto, incluyendo:Nuestro proyecto se basa en crear un programa que, a través, del análisis de datos, pueda recomendar películas a partir de estos

* Propósito o problema que resuelve: Usualmente las personas no encuentran películas de su gusto, entonces está el típico problema en el que te quedas mucho tiempo buscando películas y no encuentras nada, entonces dejas las películas a un lado y te pones a hacer otras cosas, nuestro propósito es ayudar a que estas personas tengan varias opciones de películas, y ellos poder elegir entre estas.
* Público objetivo o aplicación práctica: Nuestro programa esta diseñado para todo tipo de público, entonces cualquier persona lo puede usar, no importa la edad, su aplicación seria en cualquier cosa en el que una persona este buscando una película, y no encuentre que ver
* Resultado esperado: Generar una ayuda a las personas para que no les tome mucho tiempo elegir una película, y poder solucionar este problema tan comun

**4. Objetivos**

**General:**

* El proposito principal es brindar una ayuda eficiente y rapida a esas personas que nunca encuentran peliculas que se acoplen a sus gustos, entonces deciden mejor no ver nada, nuestra aplicacion, busca solucionar ese problema

**Específicos:**

Implementar estructuras de datos eficientes (listas y diccionarios) para almacenar y organizar la información de las películas. Aplicar técnicas de análisis de datos para identificar similitudes entre las películas y asi mejorar la precisión de las recomendaciones. Utilizar funciones y modularización para mantener un código limpio, reutilizable y fácil de mantener**.**

* Herramientas y tecnologías utilizadas (Python, librerías, etc.): se uso el lenguaje de programacion de python, tambien se uso la libreria 'pandas' para que fuese posible hacer el uso de los archivos .csv (siendo estos credits.csv y movies\_metadata.csv), asi mismo tambien se uso la biblioteca 'sklearn' para la búsqueda de similitudes entre películas y asi poder recomendarlas, y de las ultimas librerias que se utilizo fue la libreria 'os' que es la que permite ubicar los datos con los que se trabajara, y la ultima libreria que se utilizo fue 'tkinter' que nos permitio toda la parte de la interfaz del proyecto.
* Requisitos de instalación o ejecución: para la ejecucion del programa se necesitan los archivos credits.csv y movies\_metadata.csv. Tambien se necesita tener phyton descargado con las librerias 'pandas','sklearn','ast','os' y 'tkinter'

**6. Diseño del Proyecto**

* **Arquitectura o estructura del programa:** (modularización, funciones, clases, etc.): la arquitectura del programa se basa en la programacion modular, se usaron las funciones get\_recomendations() y buscar()
* **Diagrama de flujo**

Establecer carpeta de trabajo(os.chdir) https://www.falstad.com/circuit/circuitjs.html?ctz=CQAgjCAMB0l3BWcMBMcUHYMGZIA4UA2ATmIxAUgoqoQFMBaMMAKAEkRMVwNCRtsfMLyhRoSBO04ZuKPHn6DO80TAlSu-PABZFfbDtXiKLADI8hckMUsKqVAGYBDADYBnOtUhmLnBHxs-PnsQZ3dPJG9zYX1CBUDBO1Ewjy8WMEotXTQFbWxZfFFsUUkMqgNslTzZFXKS9MyK-jiQauak7loG8sNtbVz81v7RTpMyrP5uNsGqbnVxpuYpwaWR+oB3XzArTRioFk29lH9pGrsD30TTie9DkW2FXatb32O+TTf9u9jHmXavvSXJRgYIXARCfSGEEA8HKBSw4YvBHw3rnTZPX7cQIvTQJQzYsFKJqwh4w4FWRbPC64-QILGgzbYOnWWncMDETpg5ns7iLDlk94qJlnGHMuTwsWFJGSlHZKWEyG6YU3Lm8wzKlAoXTS2Raiaa7VgtCXFp7JGmkRXTTmzH6v44v6fK0nG1Bf7imEtD1ND0O3ktK5XV2BgOQQ2M0NK9VhgE0-4EiPxH78ZnB5nE1NggPZ2We3PEwzBnOtGZZgZq3RtF7TKaEJWczZtVbaOvgbb7DJzdut-xUVuzailZkIdta2iEWbaEKjIddqZwCgT1oLgddRut5utvD2C4t3Tbijtg8vEdTXKtxG7i-n+sAto5Vqth-Vp+FPecPUv7J608f8OH2QTnfT5q0GD1gNqK9KhvN1QP9GCgyg-53yaL8Jl-VCLl-bRoXfHCGUfStcIvS910rYY2j6c4AA9WgPTUKGNY5Wk4XQACcAHsACsOJYWj+jZQhyH8IQMCobJdCcABbJw2IASxcFxeP4vBigECQ8kUXJWJAAA3Og2IAEzodJCDmJdWzM3QnxAYznAAVxcAAXBgXDoQzwFUMRIFYEFzMnPs9THThbLoBznNc9zPJCGAfNM-zl0CysF24OynEcly3I8iAYtgVggA

Cargar archivos CSV

Movies\_metadata.csv

Credits.csv

Limpiar y unir datos por id

Convertir id a numérico

Rellenar NaN en overview/title

Vectorizar “overview” con TF-IDF

TfidfVectorizer(stop\_words=”english”)

Si no hay: Mostrar “Sin resultados”

Si hay: Obtener top 5 similares

Buscar coincidencias

Mostar recomendaciones(título+vote\_average)

Interfaz Tzkinter: usuario ingresa título

Crear índices por título (búsqueda)

Calcular similitud

* **Interfaz (si aplica):** Interfaz de usuario gráfica, Aplicación

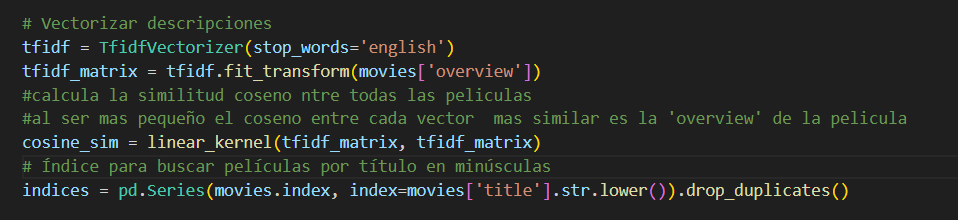
  El contenido generado por IA puede ser incorrecto.

**7. Desarrollo**

* Explicación paso a paso de cómo se desarrolló el proyecto

El primer paso fue seleccionar el proyecto atravez de la pagina: [Más de 60 proyectos Python para todos los niveles de experiencia | DataCamp](https://www.datacamp.com/es/blog/60-python-projects-for-all-levels-expertise) de la cual seleccionamos el proyecto que trata sobre un sistema de recomendación de películas. A partir de ahí empezamos a buscar que necesitábamos para nuestro proyecto, y la primera gran necesidad que tuvimos fue encontrar una base de datos para tener que películas recomendar y su puntaje (y futuramente recomendar películas en base a su reparto), una vez encontrada la base de datos nombrada en el link siendo esta (movies\_metadata.csv) buscamos como interactuar con esta. y gracias a las librerías: pandas, os y ast logramos hacer el programa capaz de encontrar el archivo .csv leerlo y interpretarlo y así al interpretar la base de datos como un vector porfin podemos empezar a buscar similitudes de las películas a partir de su resumen o ‘overview’ encontrando palabras importantes y iguales como ‘superheroes’ o ‘romance’ y al encontrar estas palabras comparamos la calificación de las películas con similitudes encontradas y seleccionamos las 5 con puntuación mas alta. Y así ya con el procedimiento claro empezamos a buscar como podíamos generar una interfaz a nuestro proyecto para que no solo corriera en la terminal y así dimos con la biblioteca tkinter

* Fragmentos de código relevantes comentados



Texto

El contenido generado por IA puede ser incorrecto.

* Descripción de las funciones principales

get\_recommendations(title): Devuelve las 5 películas más similares al título dado, junto con su calificación promedio

buscar(): Toma el texto ingresado por el usuario, busca coincidencias en títulos y muestra recomendaciones en la interfaz.

**8. Pruebas y Resultados**

* Cómo se probó el programa
* Capturas de pantalla o ejemplos de ejecución
* Resultados obtenidos
* Manual de usuario

**9. Conclusiones**

* Lecciones aprendidas
* Dificultades encontradas y cómo se resolvieron
* Posibles mejoras o ideas futuras

**10. Bibliografía / Recursos**

* [Sistemas de Recomendación en Python: Motores de recomendación basados en el contenido y el filtrado colaborativo | DataCamp](https://www.datacamp.com/es/tutorial/recommender-systems-python)
* [Full TMDB Movies Dataset 2024 (1M Movies)](https://www.kaggle.com/datasets/asaniczka/tmdb-movies-dataset-2023-930k-movies)