

Reporte Proyecto Final

Fundamentos de Programación

Diego Couto

Concepto:

- Gestor de medios de entretenimiento.

Propósito:

- Gestionar los medios de entretenimiento del usuario como: Películas y series por ver, películas y series vistas, juegos por jugar y juegos jugados.

Funcionamiento:

- Menú principal:

```
[null@archlinux proyecto_final]$ python gestor_medios.py
#####
# Gestor de Entretenimiento #
#####

Menú principal:
1. Mostrar obras
2. Ordenar obras alfabéticamente
3. Registrar obra por ver/jugar
4. Registrar obra vista
5. Sugerir obra por ver aleatoria
6. Salir

¿Qué opcion elige?:
```

- Función 1. Mostrar obras:

```
¿Qué opcion elige?: 1

1. Película
2. Serie
3. Juego
4. Libros (en desarrollo)
5. Salir

Tipo de medio (1-3): 1

1. Mostrar películas vistas
2. Mostrar películas por ver
3. Mostrar ambas
4. Salir

¿Qué películas mostrar? (1-3): 3
```

¿Qué películas mostrar? (1-3): 3

Películas por ver:

Película: Dune Razón: I read the book
 Película: Dune 2 Razón: When i read the book
 Película: Ghibli films Razón: Pop culture
 Película: Lord of the rings Razón: Pop culture
 Película: March comes in like a lion Razón: Recomendation
 Película: The hobbit Razón: Pop culture
 Película: The matrix Razón: Pop culture
 Película: The perks of being a wall flower Razón: Recomendation
 Película: Weathering with you Razón: Amazing premise

Películas vistas:

Película: A silent voice Razón: The heft of the characters emotions
 Película: Baby driver Razón: Chase scenes
 Película: Batman the dark knight Razón: The jokers character
 Película: Blade runner 2042 Razón: The cinematography
 Película: I am mother Razón: Mothers character
 Película: Inside Razón: The acting
 Película: Iron man Razón: The suits and shit
 Película: Nichts neues im westen Razón: The horrors of war
 Película: Rush Razón: Niki laudas recovery
 Película: Suzume Razón: 1road trip in the convertible
 Película: The godfather Razón: The sopranos way of life
 Película: The pianist Razón: Survival parthero 6

¿Mostrar otras listas? (s/n): n

- Función 2. Ordenar Alfabéticamente

Menú principal:

1. Mostrar obras
2. Ordenar obras alfabéticamente
3. Registrar obra por ver/jugar
4. Registrar obra vista
5. Sugerir obra por ver aleatoria
6. Salir

¿Qué opcion elige?: 2

Ordenando medios alfabeticamente...

'games_played.txt' ha sido ordenado alfabeticamente.
 'games_to_play.txt' ha sido ordenado alfabeticamente.
 'movies_to_watch.txt' ha sido ordenado alfabeticamente.
 'movies_watched.txt' ha sido ordenado alfabeticamente.
 'shows_to_watch.txt' ha sido ordenado alfabeticamente.
 'shows_watched.txt' ha sido ordenado alfabeticamente.

- Función 3. Registrar obra por ver/jugar

```

Menú principal:
1. Mostrar obras
2. Ordenar obras alfabéticamente
3. Registrar obra por ver/jugar
4. Registrar obra vista
5. Sugerir obra por ver aleatoria
6. Salir

¿Qué opcion elige?: 3

1. Película
2. Serie
3. Juego
4. Libro (en desarrollo)
5. Salir

Tipo de medio (1-3): 2

Título de Serie: la rosa de guadalupe

Razón para ver/jugar: para tirarle hate no c

Serie agregad@ con éxito :)

¿Agregar otra obra? (s/n): n

```

- Función 4. Registrar obra vista

```

Menú principal:
1. Mostrar obras
2. Ordenar obras alfabéticamente
3. Registrar obra por ver/jugar
4. Registrar obra vista
5. Sugerir obra por ver aleatoria
6. Salir

¿Qué opcion elige?: 4

1. Película
2. Serie
3. Juego
4. Salir

Tipo de medio (1-3): 2

Título de Serie: la rosa de guadalupe

Parte favorita: null

```

1. Me encantó
2. Me gustó
3. No me gustó

Gusto?: 1

Serie agregada con éxito :)

¿Agregar otra obra a vistos? (s/n): n

- Función 5. Sugerir obra por ver aleatoria

Menú principal:

1. Mostrar obras
2. Ordenar obras alfabéticamente
3. Registrar obra por ver/jugar
4. Registrar obra vista
5. Sugerir obra por ver aleatoria
6. Salir

¿Qué opcion elige?: 5

1. Película
2. Serie
3. Juego
4. Salir

Tipo de medio (1-3): 3

Sugerencia: Mouthwashing

Razón: Interesting style

¿Otra sugerencia? (s/n): s

Sugerencia: Outer wilds

Razón: Recommendation

¿Otra sugerencia? (s/n): n

- 6. Salir del programa

```

Menú principal:
1. Mostrar obras
2. Ordenar obras alfabéticamente
3. Registrar obra por ver/jugar
4. Registrar obra vista
5. Sugerir obra por ver aleatoria
6. Salir

¿Qué opcion elige?: 6

Saliendo...
[null@archlinux proyecto_final]$

```

Código Fuente:

```
"""
```

Este programa gestiona listas de medios (películas, series, juegos) del usuario. Permite consultar registros, agregar medios vistos, sugerir aleatoriamente medios por ver, y mantener archivos separados para vistos y por ver.

```
"""
```

```

import random
print('\033[32m#####')
print('# Gestor de Entretenimiento #')
print('#####\033[0m')

```

```

#\033[31m \033[0m → red
#\033[32m \033[0m → green
#\033[33m \033[0m → yellow
#\033[34m \033[0m → blue
#\033[35m \033[0m → magenta
#\033[36m \033[0m → cyan
#\033[0m → reset (always use at the end)

```

```
#funciones de cada opción
```

```
#1. consultar registros
```

```
def consult_reg():
```

```
    """
```

```
        consultar registros
```

```
    """
```

```
    #loop to read files
```

```
    b = 1
```

```
    while b == 1:
```

```
        print('\n1. Película')
```

```
        print('2. Serie')
```

```
        print('3. Juego')
```

```
        print('4. Libros (en desarrollo)')
```

```

print('5. Salir')
elec = input("\n\033[33mTipo de medio (1-3): \033[0m'

#case salir
if elec == '5':
    b = 0
#case movies
if elec == '1':
    #loop to select watched and to watch
    c = 1
    while c ==1:
        print('\n1. Mostrar películas vistas')
        print('2. Mostrar películas por ver')
        print('3. Mostrar ambas')
        print('4. Salir')
        elec_b = input("\n\033[33m¿Qué películas mostrar? (1-3): \033[0m'

#case movies watched
if elec_b == '1':
    #end loop c
    c = 0
    print('\033[32m\nPelículas vistas:\033[0m')
    try:
        with open("movies_watched.txt","r", encoding="utf-8") as archivo:
            lines = archivo.readlines()
            for x in lines:
                line = x.strip().split(',')
                print(f'\033[34mPelícula:\033[0m {line[0].capitalize()} \033[34mRazón:\033[0m
{line[1].capitalize()}')
    except:
        print('\033[31m\nError: Archivo faltante: movies_watched.txt \033[0m')
        sn = input("\033[33m\n¿Mostrar otras listas? (s/n): \033[0m')
        if sn in ('n', 'N'):
            b = 0

#case movies to watch
if elec_b == '2':
    #end loop c
    c = 0
    print('\033[32m\nPelículas por ver:\033[0m')
    try:
        with open("movies_to_watch.txt","r", encoding="utf-8") as archivo:
            lines = archivo.readlines()
            for x in lines:

```

```

        line = x.strip().split(',')
        print(f'\033[34mPelícula:\033[0m {line[0].capitalize()} \033[34mRazón:\033[0m
{line[1].capitalize()}')
    except:
        print('\033[31m\nError: Archivo faltante: movies_to_watched.txt \033[0m')
        sn = input('\033[33m\n¿Mostrar otras listas? (s/n): \033[0m')
        if sn in ('n', 'N'):
            b = 0

```

```

#case show both movies to watch and watched movies
if elec_b == '3':
    #end loop c
    c = 0
    try:
        print('\033[32m\nPelículas por ver:\033[0m')
        with open("movies_to_watch.txt", "r", encoding="utf-8") as archivo:
            lines = archivo.readlines()
            for x in lines:
                line = x.strip().split(',')
                print(f'\033[34mPelícula:\033[0m {line[0].capitalize()} \033[34mRazón:\033[0m
{line[1].capitalize()}')
    except:
        print('\033[31m\nError: Archivo faltante: movies_to_watched.txt \033[0m')
        print('\033[32m\nPelículas vistas:\033[0m')
        try:
            with open("movies_watched.txt", "r", encoding="utf-8") as archivo:
                lines = archivo.readlines()
                for x in lines:
                    line = x.strip().split(',')
                    print(f'\033[34mPelícula:\033[0m {line[0].capitalize()} \033[34mRazón:\033[0m
{line[1].capitalize()}')
        except:
            print('\033[31m\nError: Archivo faltante: movies_watched.txt \033[0m')
            sn = input('\033[33m\n¿Mostrar otras listas? (s/n): \033[0m')
            if sn in ('n', 'N'):
                b = 0

```

```

#case series
if elec == '2':
    #loop to select watched and to watch
    c = 1
    while c == 1:
        print('\n1. Mostrar series vistas')
        print('2. Mostrar series por ver')

```

```

print('3. Mostrar ambas')
elec_b = input("\n¿Qué series mostrar? (1-3): ")

if elec_b == '1':
    #end loop c
    c = 0
    print("\nSeries vistas:")
    with open("shows_watched.txt", "r", encoding="utf-8") as archivo:
        lines = archivo.readlines()
    for x in lines:
        line = x.strip().split(',')
        print(f"Series: {line[0].capitalize()} Razón: {line[1].capitalize()}")
    sn = input("\n¿Mostrar otras listas? (s/n): ")
    if sn in ('n', 'N'):
        b = 0

if elec_b == '2':
    #end loop c
    c = 0
    print("\nSeries por ver:")
    with open("shows_to_watch.txt", "r", encoding="utf-8") as archivo:
        lines = archivo.readlines()
    for x in lines:
        line = x.strip().split(',')
        print(f"Series: {line[0].capitalize()} Razón: {line[1].capitalize()}")
    sn = input("\n¿Mostrar otras listas? (s/n): ")
    if sn in ('n', 'N'):
        b = 0

if elec_b == '3':
    #end loop c
    c = 0
    print("\nSeries por ver:")
    with open("shows_to_watch.txt", "r", encoding="utf-8") as archivo:
        lines = archivo.readlines()
    for x in lines:
        line = x.strip().split(',')
        print(f"Series: {line[0].capitalize()} Razón: {line[1].capitalize()}")
    print("\nSeries vistas:")
    with open("shows_watched.txt", "r", encoding="utf-8") as archivo:
        lines = archivo.readlines()

```

```

    for x in lines:
        line = x.strip().split(',')
        print(f'\033[34mSerie:\033[0m {line[0].capitalize()} \033[34mRazón:\033[0m
{line[1].capitalize()}')
        sn = input('\033[33m\n¿Mostrar otras listas? (s/n): \033[0m')
        if sn in ('n', 'N'):
            b = 0

#case juegos
if elec == '3':
    #loop to select watched and to watch
    c = 1
    while c == 1:
        print('\n1. Mostrar juegos jugados')
        print('2. Mostrar juegos por jugar')
        print('3. Mostrar ambos')
        elec_b = input('\n\033[33m¿Qué juegos mostrar? (1-3): \033[0m')

        if elec_b == '1':
            #end loop c
            c = 0
            print('\033[32m\nJuegos jugados:\033[0m')
            with open("games_played.txt", "r", encoding="utf-8") as archivo:
                lines = archivo.readlines()
                for x in lines:
                    line = x.strip().split(',')
                    print(f'\033[34mJuego:\033[0m {line[0].capitalize()} \033[34mRazón:\033[0m
{line[1].capitalize()}')
                    sn = input('\033[33m\n¿Mostrar otras listas? (s/n): \033[0m')
                    if sn in ('n', 'N'):
                        b = 0

        if elec_b == '2':
            #end loop c
            c = 0
            print('\033[32m\nJuegos por jugar:\033[0m')
            with open("games_to_play.txt", "r", encoding="utf-8") as archivo:
                lines = archivo.readlines()
                for x in lines:
                    line = x.strip().split(',')
                    print(f'\033[34mJuego:\033[0m {line[0].capitalize()} \033[34mRazón:\033[0m
{line[1].capitalize()}')
                    sn = input('\033[33m\n¿Mostrar otras listas? (s/n): \033[0m')
                    if sn in ('n', 'N'):

```

```

b = 0

if elec_b == '3':
    #end loop c
    c = 0
    print('\033[32m\nJuegos por jugar:\033[0m')
    with open("games_to_play.txt", "r", encoding="utf-8") as archivo:
        lines = archivo.readlines()
    for x in lines:
        line = x.strip().split(',')
        print(f'\033[34mJuego:\033[0m {line[0].capitalize()} \033[34mRazón:\033[0m
{line[1].capitalize()}')
        print('\033[32m\nJuegos jugados:\033[0m')
        with open("games_played.txt", "r", encoding="utf-8") as archivo:
            lines = archivo.readlines()
        for x in lines:
            line = x.strip().split(',')
            print(f'\033[34mJuego:\033[0m {line[0].capitalize()} \033[34mRazón:\033[0m
{line[1].capitalize()}')
        sn = input('\033[33m\n¿Mostrar otras listas? (s/n): \033[0m')
        if sn in ('n', 'N'):
            b = 0

#rewrite files in alphabetical order
def ordenar_medios():
    """
    rewrite files in alphabetical order
    """

    def sort_name_file(filename):
        # open file and read all lines
        with open(filename, "r", encoding="utf-8") as archivo:
            lines = archivo.readlines()
        #sort lines by name (before split,"")
        lines.sort(key=lambda line: line.split(",")[0].strip().lower())
        #rewrite the file with sorted lines
        with open(filename, "w", encoding="utf-8") as archivo:
            archivo.writelines(lines)
        print(f'\033[32m'{filename}' ha sido ordenado alfabeticamente. \033[0m")

    #sort each txt file
    sort_name_file("games_played.txt")
    sort_name_file("games_to_play.txt")
    sort_name_file("movies_to_watch.txt")
    sort_name_file("movies_watched.txt")
    sort_name_file("shows_to_watch.txt")
    sort_name_file("shows_watched.txt")

```

#2. registrar nueva serie/película/juego por jugar/ver

```
def add_pice(media_type,file):
    """
    función add media to media to watch
    """
    titulo = input(f'\033[33m\nTítulo de {media_type}: \033[0m')
    titulo = titulo.lower()
    razon = input(f'\033[33m\nRazón para ver/jugar: \033[0m')
    razon = razon.lower()
    with open(file,"a+", encoding="utf-8") as archivo:
        archivo.write(f'{titulo},{razon}\n')
    print(f'\033[32m\n{media_type} agregad@ con éxito :) \033[0m')
```

```
def add_por_ver():
    """
    loop to add media to their respective files
    """
    b = 1
    while b == 1:
        print('\n1. Película')
        print('2. Serie')
        print('3. Juego')
        print('4. Libro (en desarrollo)')
        print('5. Salir')
        elec = input('\n\033[33mTipo de medio (1-3): \033[0m')
        #case salir
        if elec == '5':
            b = 0
        #case add movie
        if elec == '1':
            add_pice("Película","movies_to_watch.txt")
            sn = input(f'\033[33m\n¿Agregar otra obra? (s/n): \033[0m')
            if sn in ('n', 'N'):
                #exit loop
                b = 0

        #case add show
        if elec == '2':
            add_pice("Serie","shows_to_watch.txt")
            sn = input(f'\033[33m\n¿Agregar otra obra? (s/n): \033[0m')
            if sn in ('n', 'N'):
                #exit loop
                b = 0
```

```

#case add game
if elec == '3':
    add_pice("Game","games_to_play.txt")
    sn = input("\033[33m\n¿Agregar otra obra? (s/n): \033[0m'")
    if sn in ('n', 'N'):
        #exit loop
        b = 0

```

```

#3. registrar medio visto, si está en medios por ver, eliminar
def add_medio_visto_tipo(media_type,towatch_file,watched_file):
    """
    3. registrar medio visto, si está en medios por ver, eliminar
    """
    titulo = input(f'\033[33m\nTítulo de {media_type}: \033[0m')
    titulo = titulo.lower()
    parte_fav = input('\033[33m\nParte favorita: \033[0m')
    #bucle set score
    e = 1
    while e == 1:
        score = int(input("\n1. Me encantó\n2. Me gustó\n3. No me gustó\n\n\033[33mGusto?: \033[0m'))
        if 0 < score < 4:
            e = 0
    # quitar película de películas por ver
    with open(towatch_file, "r", encoding="utf-8") as archivo:
        lines = archivo.readlines()
    line_to_remove = None
    for line in lines:
        # split line at ','
        arr = line.strip().split(",")
        # compare titles (case-insensitive)
        if arr[0].lower() == titulo:
            line_to_remove = line
            # stop after first match
            break
    #if line_to_remove is not empty, rewrite file without line_to_remove
    if line_to_remove:
        with open(towatch_file, "w", encoding="utf-8") as archivo:
            for line in lines:
                if line != line_to_remove:
                    archivo.write(line)
    #check if title already exists in movies_watched.txt
    already_existst = 0

```

```

with open(watched_file, "r", encoding="utf-8") as archivo:
    lines = archivo.readlines()
line_to_remove = None
#check line by line:
for line in lines:
    # split line at ','
    arr = line.strip().split(",")
    # compare titles (case-insensitive)
    if arr[0].lower() == titulo:
        already_existst = 1
#agregar película a películas vistas
if already_existst == 0:
    with open(watched_file,"a+", encoding="utf-8") as archivo:
        archivo.write(f"{titulo},{parte_fav},{score}\n")
    print(f"\033[32m\n{media_type} agregada con éxito :) \033[0m')

#actual function that is called from the main menu
def reg_medio_visto():
    """
    actual function that is called from the main menu
    """
    c = 1
    while c == 1:
        print('\n1. Película')
        print('2. Serie')
        print('3. Juego')
        print('4. Salir')
        elec = input("\n\033[33mTipo de medio (1-3): \033[0m')

        #case salir
        if elec == '4':
            c = 0
        #case add movie
        if elec == '1':
            add_medio_visto_tipo("Película","movies_to_watch.txt","movies_watched.txt")
            sn = input("\033[33m\n¿Agregar otra obra a vistos? (s/n): \033[0m')
            if sn in ('n', 'N'):
                #exit loop
                c = 0

        #case add show
        if elec == '2':
            add_medio_visto_tipo("Serie","shows_to_watch.txt","shows_watched.txt")
            sn = input("\033[33m\n¿Agregar otra obra a vistos? (s/n): \033[0m')

```

```

if sn in ('n', 'N'):
    #exit loop
    c = 0

#case add show
if elec == '3':
    add_medio_visto_tipo("Juego","games_to_play.txt","games_played.txt")
    sn = input('\033[33m\n¿Agregar otra obra a vistos? (s/n): \033[0m')
    if sn in ('n', 'N'):
        #exit loop
        c = 0

```

#5. suggest random towatch peace of media

def sugest_ran_towatch():

```

'''
5. suggest random towatch peace of media
'''

c = 1
while c == 1:
    print('\n1. Película')
    print('2. Serie')
    print('3. Juego')
    print('4. Salir')
    elec = input('\n\033[33mTipo de medio (1-3): \033[0m')

    #case salir
    if elec == '4':
        c = 0
    #case movie
    if elec == '1':
        d = 1
        while d == 1:
            with open("movies_to_watch.txt","r", encoding="utf-8") as archivo:
                lines = archivo.readlines()
            #generate random number between the values of 1 and the number of lines
            num = random.randint(0, len(lines)-1)
            #read line corresponding to the random number as an index
            line = lines[num].split(',')
            print(f'\033[34m\nSugerencia:\033[0m {line[0].capitalize()}')
            print(f'\033[34mRazón:\033[0m {line[1].capitalize()}')
            #exit loop
            sn = input('\n\033[33m¿Otra sugerencia? (s/n): \033[0m')
            if sn in ('n', 'N'):
                #exit loop

```

```

    d = 0
    c = 0

```

```

#case show

```

```

if elec == '2':

```

```

    d = 1

```

```

    while d == 1:

```

```

        with open("shows_to_watch.txt", "r", encoding="utf-8") as archivo:

```

```

            lines = archivo.readlines()

```

```

            #generate random number between the values of 1 and the number of lines

```

```

            num = random.randint(0, len(lines)-1)

```

```

            #read line corresponding to the random number as an index

```

```

            line = lines[num].split(',')

```

```

            print(f'\033[34m\nSugerencia:\033[0m {line[0].capitalize()}')

```

```

            print(f'\033[34mRazón:\033[0m {line[1].capitalize()}')

```

```

            #exit loop

```

```

            sn = input('\n\033[33m¿Otra sugerencia? (s/n): \033[0m')

```

```

            if sn in ('n', 'N'):

```

```

                #exit loop

```

```

                d = 0

```

```

                c = 0

```

```

#case movie

```

```

if elec == '3':

```

```

    d = 1

```

```

    while d == 1:

```

```

        with open("games_to_play.txt", "r", encoding="utf-8") as archivo:

```

```

            lines = archivo.readlines()

```

```

            #generate random number between the values of 1 and the number of lines

```

```

            num = random.randint(0, len(lines)-1)

```

```

            #read line corresponding to the random number as an index

```

```

            line = lines[num].split(',')

```

```

            print(f'\033[34m\nSugerencia:\033[0m {line[0].capitalize()}')

```

```

            print(f'\033[34mRazón:\033[0m {line[1].capitalize()}')

```

```

            #exit loop

```

```

            sn = input('\n\033[33m¿Otra sugerencia? (s/n): \033[0m')

```

```

            if sn in ('n', 'N'):

```

```

                #exit loop

```

```

                d = 0

```

```

                c = 0

```

```

#menú principal

```

```

A = 1

```

```

while A == 1:

```

```
print('\n\033[34mMenú principal: \033[0m')
print('1. Mostrar obras')
print('2. Ordenar obras alfabéticamente')
print('3. Registrar obra por ver/jugar')
print('4. Registrar obra vista')
print('5. Sugerir obra por ver aleatoria')
print('6. Salir')
#try:
eleccion = input('\033[33m\n¿Qué opcion elige?: \033[0m')
#caso 1
if eleccion == '1':
    consult_reg()
#caso 2
if eleccion == '2':
    print('Ordenando medios alfabeticamente...')
    ordenar_medios()
#caso 3
if eleccion == '3':
    add_por_ver()
#caso 4
if eleccion == '4':
    reg_medio_visto()

#caso 5
if eleccion == '5':
    sugest_ran_towatch()

#caso 6
if eleccion == '6':
    print('\033[32m\nSaliendo...\033[0m')
    A = 0
```