

▼ ABP4_Teledocencia

▼ Miembros

Alejandro Gómez González

Miguel Arias Pérez

Juan Carlos Fernandes González

Juan José Poza Mariño

▼ Preparación

Aquí se muestran las ejecuciones necesarias para inicializar el entorno de trabajo

Manual de uso

En esta sección se cargan y muestran los archivos de datos necesarios para la aplicación, los siguientes archivos CSV son leídos y almacenados en un DataFrame:

-seriesData: Almacena toda la información de las series (descripción, año, título, etc...)

-gustosData: Almacena la información de los gustos de cada usuario, indicando el id del usuario y los tipos de géneros que le gustan.

-series_visData: Almacena la información de las series que ha visualizado cada usuario, indicando el id del usuario, el id de la serie y su grado de satisfacción con esa serie, puede tomar tres valores:

-1: No le ha gustado

0: Indiferente

1: Le ha gustado mucho

-usuarioData: Almacena la información de cada usuario (DNI,login,password,etc...).

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
```

```

seriesData = pd.read_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/series.c
series_visData = pd.read_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/Seri
gustosData = pd.read_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/Gustos_
usuarioData = pd.read_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/usuaric
 analisisData = pd.read_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/Final.
series_comentariosData = pd.read_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/

```

seriesData

	Actores	Director	Genero	Descripcion	Anho	Titulo
0	Rami Malek, Christian Slater, Carly Chaikin, M...	Sam Esmail	Crime, Drama, Thriller	Elliot, a brilliant but highly unstable young ...	24 Jun 2015	Mr. Robot
1	Cillian Murphy, Paul Anderson, Helen McCrory, ...	NaN	Crime, Drama	A gangster family epic set in 1919 Birmingham,...	30 Sep 2014	Peaky Blinders
2	Karoline Eichhorn, Louis Hofmann, Jödis Trieb...	NaN	Crime, Drama, Mystery, Sci-Fi, Thriller	A family saga with a supernatural twist, set i...	01 Dec 2017	Dark
3	Bryan Cranston, Anna Gunn, Aaron Paul, Betsy B...	NaN	Crime, Drama, Thriller	A high school chemistry teacher diagnosed with...	20 Jan 2008	Breaking Bad
4	Peter Dinklage, Lena Headey, Emilia Clarke, Ki...	NaN	Action, Adventure, Drama, Fantasy, Romance	Nine noble families fight for control over the...	17 Apr 2011	Game of Thrones
...

usuarioData

	DNI	Login	Password	Email	Telefono
0	72454396R	bschulken0	2bMlzAB32yp	lklimentyonok0@cdbaby.com	165707677
1	99029032W	mkyteley1	6baXWmYIGVg	fwreak1@instagram.com	453162970
2	66878252D	khaynesford2	fB4vUiHOOE	bstonall2@walmart.com	415965212
3	68378360B	tbarrand3	kuR6o4I	askett3@usnews.com	710920310
4	86351111L	ajiruca4	mwEDUTFJHR	djahnel4@microsoft.com	765272967
...
95	34536921D	cfitzgerald2n	7rJ5n37Xj	mavramov2n@prweb.com	843236117
96	03284481R	stregunna2o	v0gUWwLF8VM	ekivelle2o@moonfruit.com	262281420
97	87746132S	tnickoll2p	ppyA2TNI1	dpirdy2p@com.com	815085588
98	42597010B	dlukianov2q	YbXS6csIPd	afeather2q@illinois.edu	800356764
99	41282090S	apriscott2r	Dm6zay	abreazeall2r@amazon.de	29013525

100 rows × 5 columns

series_visData

	DNI	Titulo	Satisfaccion
0	41282090S	Monk	negative
1	41282090S	Dark	positive
2	41282090S	Psych	positive
3	41282090S	Breaking Bad	negative
4	41282090S	White Collar	negative
5	68378360B	Dark	positive

gustosData

	DNI	Genero
0	24578169L	Adventure
1	24578169L	Thriller
2	24578169L	Comedy
3	45144479X	Comedia
4	41282090S	Comedia
5	41282090S	fantasia

analisisData #Aprendizaje

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The filming tec...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's Love in the Time of Money is a...	positive
...
994	On watching this film, I was amazed at how med...	positive
995	Nothing is sacred. Just ask Ernie Fosselius. T...	positive
996	I hated it. I hate self-aware pretentious inan...	negative
997	I usually try to be professional and construct...	negative
998	If you like me is going to see this in a film ...	negative

999 rows × 2 columns

series_comentariosData

	DNI	Titulo	Comentario	Valoracion
0	72454396R	Dark	Really good serie	positive
1	68378360B	Monk	Quite enjoyable	positive
2	45144479X	Dark	I was amazed about the good plot and best acto...	['positive']
3	41282090S	Monk	not very pleasant to see	negative
4	41282090S	Dark	the film was a bit boring with no many action ...	['positive']
5	41282090S	The Walking Dead	The start was boring with no action, but in th...	['negative']
6	41282090S	The Simpsons	One of the most entertaining serie i have ever...	['positive']
7	41282090S	Dark	Good serie	[negative]
8	41282090S	The walking dead	good serie with great actors and nice music	[positive]

▼ Gestionar Series

A continuación se detallan los casos de uso de añadir, modificar, consultar y eliminar series.

▼ Añadir Serie

Para añadir una serie, se muestran los campos a rellenar para insertar una nueva serie, y se presiona el botón Añadir.

Manual de uso: Añadir Serie

Para añadir una serie es necesario ejecutar la función añadirSerie(), esta función solicitará que se introduzcan los valores de una serie:

1. Se introduce el título de la serie
2. Se introduce la descripción de la serie
3. Se introduce el género de la serie
4. Se introduce el año de la serie
5. Se introducen los actores de la serie

Una vez introducidos todos los parámetros, el sistema inserta la serie en el DataFrame seriesData, la ejecución de la función finaliza devolviendo los parámetros de la serie que se

```
def añadirSerie():
    Titulo = input ("Por favor, introduce el título de la serie: ")
    Descripcion = input ("Por favor, introduce la descripción de la serie: ")
    Genero = input ("Por favor, introduce el género de la serie: ")
    Anho = input ("Por favor, introduce el año de la serie: ")
    Actores = input ("Por favor, introduce los actores de la serie: ")
    Director = input ("Por favor, introduce el director de la serie: ")

    seriesData.loc[seriesData.shape[0]] = [Actores,Director,Genero,Descripcion,Anho,Titulo]
    seriesData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/series.csv')
    print ("-----")
    print (" ¡Serie añadida con éxito!")
    print ("-----")
    return seriesData.tail(1)
añadirSerie()
```

```
Por favor, introduce el título de la serie: Prueba
Por favor, introduce la descripción de la serie: Probando
Por favor, introduce el género de la serie: Comedia
Por favor, introduce el año de la serie: 2020
Por favor, introduce los actores de la serie: A
Por favor, introduce el director de la serie: B
```

```
-----
¡Serie añadida con éxito!
-----
```

	id_Series	Actores	Director	Genero	Descripcion	Anho	Titulo
98	99	A	B	Comedia	Probando	2020	Prueba

▼ Consultar Serie

Para consultar una serie, se muestran las series, se escoge una y se presiona el botón Consultar.

Manual de uso: Consultar Serie

Para consultar una serie es necesario ejecutar la función consultarSerie(), esta función solicitará el título de la serie que se quiere consultar.

1. Se introduce el título de la serie que se quiere consultar

```
def consultarSerie():
    Titulo = input ("Por favor, introduce el título de la serie que desea consultar: ")
    Series = seriesData[seriesData["Titulo"] == Titulo]

    return Series
consultarSerie()
```

Por favor, introduce el título de la serie que desea consultar: Prueba

id_Series	Actores	Director	Genero	Descripcion	Anho	Titulo
98	99	A	B	Comedia	Probando	2020 Prueba

▼ Modificar Serie

Para modificar una serie, se muestran las series, se escoge una y se presiona el botón Modificar.

En el caso de que no se escriba nada se dejará el texto que tenía antes.

Manual de uso: Modificar Serie

Para modificar una serie es necesario ejecutar la función modificarSerie(), esta función solicitará el título de la serie que se quiere modificar y los nuevos valores de una serie, si no se quiere modificar un determinado valor simplemente se deja en blanco ese campo:

1. Se introduce el título de la serie que se quiere modificar
2. Se introduce el nuevo título de la serie
3. Se introduce la nueva descripción de la serie
4. Se introduce el nuevo género de la serie
5. Se introduce el nuevo año de la serie
6. Se introducen los nuevos actores de la serie

Una vez introducidos todos los parámetros, el sistema actualiza los datos de la serie en el DataFrame seriesData, la ejecución de la función finaliza devolviendo los parámetros de la serie que se acaba de actualizar.

```
def modificarSerie():
    Titulo = input ("Por favor, introduce el titulo de la serie que desea modificar: ")
    Series = seriesData[seriesData["Titulo"] == Titulo]
    valor = Series.index.values
```

```

Titulo = input ("Escriba algo si desea modificar el título de la serie: ")
if Titulo == "" :
    Titulo = Series["Titulo"].values[0]

Descripcion = input ("Escriba algo si desea modificar la descripción de la serie: ")
if Descripcion == "" :
    Descripcion = Series["Descripcion"].values[0]

Genero = input ("Escriba algo si desea modificar el género de la serie: ")
if Genero == "" :
    Genero = Series["Genero"].values[0]

Director = input ("Escriba algo si desea modificar el director de la serie: ")
if Director == "" :
    Director = Series["Director"].values[0]

Anho = input ("Escriba algo si desea modificar el año de la serie: ")
if Anho == "" :
    Anho = Series["Anho"].values[0]

Actores= input ("Escriba algo si desea modificar los actores de la serie: ")
if Actores == "" :
    Actores = Series["Actores"].values[0]

seriesData.loc[valor[0]] = [Actores,Director,Genero,Descripcion,Anho,Titulo]

seriesData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/series.csv')
print ("-----")
print (" ¡Serie modificada con éxito!")
print ("-----")
return seriesData.loc[[valor[0]]]
modificarSerie()

```

Por favor, introduce el titulo de la serie que desea modificar: Prueba
 Escriba algo si desea modificar el título de la serie:
 Escriba algo si desea modificar la descripción de la serie: Probando modificacion
 Escriba algo si desea modificar el género de la serie:
 Escriba algo si desea modificar el director de la serie:
 Escriba algo si desea modificar el año de la serie:
 Escriba algo si desea modificar los actores de la serie:

 ¡Serie modificada con éxito!

id_Series	Actores	Director	Genero	Descripcion	Anho	Titulo
98	99	A	B Comedia	Probando modificacion	2020	Prueba

▼ Eliminar Serie

Para eliminar una serie, se muestran las series, se escoge una y se presiona el botón Eliminar.

Manual de uso: Eliminar Serie

Para eliminar una serie es necesario ejecutar la función `eliminarSerie()`, esta función solicitará el título de la serie que se quiere eliminar y la borra del DataFrame `seriesData()`:

1. Se introduce el título de la serie que se quiere eliminar

Una vez eliminada la serie el sistema devuelve la lista de las series almacenadas.

```
def eliminarSerie():
    global seriesData

    Titulo = input ("Por favor, introduce el título de la serie que desea eliminar: ")
    seriesData = seriesData.drop(seriesData[seriesData['Titulo']==Titulo].index)

    seriesData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/series.csv')
    print ("-----")
    print (" ¡Serie eliminada con éxito!")
    print ("-----")
    return seriesData
```

`eliminarSerie()`

Por favor, introduce el título de la serie que desea eliminar: Pepito Grillo

 ¡Serie eliminada con éxito!

	id_Series	Actores	Director	Genero	Descripcion	Anho	Titulo
0	0	Rami Malek, Christian Slater, Carly Chaikin, M...	Sam Esmail	Crime, Drama, Thriller	Elliot, a brilliant but highly unstable young ...	24 Jun 2015	Mr. Robot
1	1	Cillian Murphy, Paul Anderson, Helen McCrory, ...	NaN	Crime, Drama	A gangster family epic set in 1919 Birmingham,...	30 Sep 2014	Peaky Blinders
2	2	Karoline Eichhorn, Louis Hofmann, Jödis Trieb...	NaN	Crime, Drama, Mystery, Sci- Fi, Thriller	A family saga with a supernatural twist, set i...	01 Dec 2017	Dark
3	3	Bryan Cranston, Anna Gunn, Aaron Paul, Betsy B...	NaN	Crime, Drama, Thriller	A high school chemistry teacher diagnosed with...	20 Jan 2008	Breaking Bad
		Peter Dinklage,		Action, Adventure	Nine noble	17	

▼ Gestionar Usuarios

▼ Añadir Usuario

Manual de uso: Añadir Usuario

Para añadir un usuario es necesario ejecutar la función `añadirUsuario()`, esta función solicitará que se introduzcan los valores del usuario así como sus gustos y las series que ha visualizado:

Se ejecuta la función **`añadirUsuario()`**

1. Se introduce el DNI del usuario
2. Se introduce el Login del usuario
3. Se introduce la contraseña del usuario
4. Se introduce el mail del usuario
5. Se introduce el teléfono del usuario

Se inserta el usuario en el DataFrame `usuarioData` y muestra la información del usuario que se acaba de introducir.

Se ejecuta la función **`añadirGusto()`**

1. Se introduce el género que le gusta al usuario

Se insertan los gustos del usuario en el DataFrame `gustosData` y se muestran los gustos del usuario que se acaba de introducir.

Se ejecuta la función **`añadirSerieVis()`**

1. Se introduce el id de la serie que ha visto el usuario
2. Se introduce el grado de satisfacción con dicha serie

Se insertan las series vistas por el usuario en el DataFrame `series_visData` y se muestran las series vistas por el usuario que se acaba de introducir.

```
def añadirUsuario():
    DNI = input ("Por favor, introduce el DNI del usuario: ")
    Login = input ("Por favor, introduce el login del usuario: ")
    Password = input ("Por favor, introduce la password del usuario: ")
    Email = input ("Por favor, introduce el email del usuario: ")
    Telefono = input ("Por favor, introduce el teléfono del usuario: ")

    usuarioData.loc[seriesData.shape[0]] = [DNI,Login>Password>Email>Telefono]
    usuarioData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/usuario.csv')
    print ("-----")
    print (" ¡Usuario añadido con éxito!")
    print (" ")
```

```
print ( ----- )
```

```
añadirGusto(DNI)
```

```
añadirSerieVis(DNI)
```

```
return usuarioData.tail(1)
```

```
def añadirGusto(DNI):
```

```
Genero = input ("Por favor, introduce un género de tu gusto: ")
```

```
gustosData.loc[gustosData.shape[0]] = [DNI,Genero]
```

```
gustosData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/Gustos_gene
```

```
print ("-----")
```

```
print (" ¡Genero añadido con éxito!")
```

```
print ("-----")
```

```
return gustosData.tail(1)
```

```
def añadirSerieVis(DNI):
```

```
Titulo = input ("Por favor, introduce el titulo de la serie que hayas visualizado: ")
```

```
Satis = input ("Por favor, introduce el grado de satisfaccion de la serie que hayas vi
```

```
series_visData.loc[series_visData.shape[0]] = [DNI,Titulo,Satis]
```

```
series_visData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/Series_
```

```
print ("-----")
```

```
print (" ¡Serie añadida con éxito!")
```

```
print ("-----")
```

```
return usuarioData.tail(1)
```

```
añadirUsuario()
```

▼ Consultar Usuarios

Para consultar un usuario, se presiona el botón Consultar en su perfil.

Manual de uso: Consultar Usuario

Para consultar un usuario es necesario ejecutar la función consultarUsuario(), esta función solicitará el DNI del usuario que se quiere consultar.

1. Se introduce el DNI del usuario que se quiere consultar

El sistema muestra la información del usuario, sus gustos y sus series vistas.

```

def consultarUsuario():
    DNI = input ("Por favor, introduce el DNI del usuario que desea consultar: ")
    Usuario = usuarioData[usuarioData["DNI"] == DNI]

    Gustos = consultarGustos(DNI)
    consultarSeriesVis(DNI)

    return Usuario

def consultarGustos(DNI):
    Gustos = gustosData[gustosData['DNI']==DNI]
    print(Gustos)

    return

def consultarSeriesVis(DNI):
    SeriesVis = series_visData[series_visData['DNI']==DNI]
    print(SeriesVis)

    return

consultarUsuario()
  
```

Por favor, introduce el DNI del usuario que desea consultar: 45144479X

DNI Genero

3 45144479X Comedia

DNI ID Satisfaccion

3 45144479X 2 0

	DNI	Login	Password	Email	Telefono
1	45144479X	Alex	Alex	jhjkm@ihnk.mnbu	867187

▼ Modificar Usuarios

Para modifciar un usuario, se presiona el botón Modificar en su perfil.

En el caso de que no se escriba nada se dejará el texto que tenía antes.

Tanto para series visualizadas como para gustos el usuario deberá decidir si quiere añadir o eliminar.

Manual de uso: Modificar Usuario

Para modificar un usuario es necesario ejecutar la función `modificarUsuario()`, esta función solicitará el DNI del usuario que se quiere modificar y los valores del usuario, en caso de no querer cambiar un valor se dejará en blanco, la función preguntará si se quieren editar los gustos del usuario y las series vistas, en caso afirmativo el sistema edita los correspondientes valores.

Se ejecuta la función **`modificarUsuario()`**

1. Se introduce el nuevo DNI del usuario
2. Se introduce el nuevo Login del usuario
3. Se introduce la nueva contraseña del usuario
4. Se introduce el nuevo mail del usuario
5. Se introduce el nuevo teléfono del usuario

Se cambian los valores del usuario en el DataFrame `usuarioData`.

El sistema pregunta si se quiere agregar o eliminar un gusto:

En caso de **Agregar**:

1. Se introduce el género que le gusta al usuario

Se insertan los gustos del usuario en el DataFrame `gustosData`.

En caso de **Eliminar**:

1. Se introduce el género que quiere eliminar del usuario

Se elimina los gustos del usuario en el DataFrame `gustosData`.

El sistema pregunta si se quiere agregar o eliminar un gusto:

En caso de **Agregar**:

1. Se introduce el id de la serie que ha visto el usuario
2. Se introduce el grado de satisfacción con dicha serie

Se insertan las series vistas por el usuario en el DataFrame `series_visData`.

En caso de **Eliminar**:

1. Se introduce el id de la serie que se quiere eliminar de las vistas por el usuario

Se elimina de las series vistas por el usuario en el DataFrame `series_visData`.

```
def modificarUsuario():
```

```
    dni = input("Por favor, introduzca el DNI del usuario que se desea modificar: ")
```

```

DNI = input ( "Por favor, introduce el DNI del usuario que se desea modificar. " )
Usuarios = usuarioData[usuarioData["DNI"] == DNI]
valor = Usuarios.index.values

Login = input ( "Escriba algo si desea modificar el login del usuario: " )
if Login == "" :
    Login = Usuarios["Login"].values[0]

Password = input ( "Escriba algo si desea modificar la password del usuario: " )
if Password == "" :
    Password = Usuarios["Password"].values[0]

Email = input ( "Escriba algo si desea modificar el email del usuario: " )
if Email == "" :
    Email = Usuarios["Email"].values[0]

Telefono = input ( "Escriba algo si desea modificar el teléfono del usuario: " )
if Telefono == "" :
    Telefono = Usuarios["Telefono"].values[0]

usuarioData.loc[valor[0]] = [DNI,Login,Password,Email,Telefono]
usuarioData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/usuario.csv')
print ( "-----" )
print ( " ¡Usuario modificado con éxito!" )
print ( "-----" )

Pregunta1 = input ( "¿Entrar en gustos?(si/no) " )
if Pregunta1 == "si" :
    PreguntaGustos = Pregunta1
    while PreguntaGustos == "si" :
        modificarGusto(DNI)
        PreguntaGustos = input ( "¿Entrar en gustos?(si?) " )

Pregunta2 = input ( "¿Entrar en series visualizadas?(si/no) " )
if Pregunta2 == "si" :
    PreguntaSeries = Pregunta2
    while PreguntaSeries == "si" :
        modificarSerieVis(DNI)
        PreguntaSeries = input ( "¿Entrar en series?(si?)" )

return usuarioData.loc[[valor[0]]]

def modificarGusto(DNI):
    global gustosData
    Pregunta = input ( "¿Quiere agregar un gusto o eliminar?" )

    if Pregunta == "agregar" :
        Genero = input ( "Por favor, introduce un género a agregar: " )
        gustosData.loc[gustosData.shape[0]] = [DNI,Genero]
        gustosData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/Gustos.csv')
        print ( "-----" )
        print ( " ¡Genero modificado con éxito!" )
        print ( "-----" )

```

```

elif Pregunta == "eliminar" :
    Genero = input ("Por favor, introduce un género a eliminar: ")
    gustosData = gustosData.drop(gustosData[(gustosData['DNI']==DNI) & (gustosData['Genero']==Genero)])
    gustosData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/Gustos_gustos.csv')
    print ("-----")
    print (" ¡Genero eliminado con éxito!")
    print ("-----")

else :
    print ("-----")
    print ("No se ha realizado ninguna modificación")
    print ("-----")

return

def modificarSerieVis(DNI):
    global series_visData

    Pregunta = input ("¿Quiere agregar serie visualizada o eliminar?")

    if Pregunta == "agregar" :
        Titulo = input ("Por favor, introduce el titulo de la serie que hayas visualizado: ")
        Satis = input ("Por favor, introduce el grado de satisfaccion de la serie que hayas visualizado: ")
        series_visData.loc[series_visData.shape[0]] = [DNI,Titulo,Satis]
        series_visData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/SerieVisualizada.csv')
        print ("-----")
        print (" ¡Serie modificada con éxito!")
        print ("-----")

    elif Pregunta == "eliminar" :
        Titulo = input ("Por favor, introduce el titulo de la serie que quieras eliminar de series_visData: ")
        series_visData = series_visData.drop(series_visData[(series_visData['DNI']==DNI) & (series_visData['Titulo']==Titulo)])
        series_visData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/SerieVisualizada.csv')
        print ("-----")
        print (" ¡Serie eliminada con éxito!")
        print ("-----")

    else :
        print ("-----")
        print ("No se ha realizado ninguna modificación")
        print ("-----")

    return

modificarUsuario()

```

```

Por favor, introduce el DNI del usuario que se desea modificar: 41282090S
Escriba algo si desea modificar el login del usuario:
Escriba algo si desea modificar la password del usuario:
Escriba algo si desea modificar el email del usuario:
Escriba algo si desea modificar el teléfono del usuario:
-----
¡Usuario modificado con éxito!
-----
¿Entrar en gustos?(si/no) no
¿Entrar en series visualizadas?(si/no) si
¿Quiere agregar serie visualizada o eliminar?agregar
Por favor, introduce el titulo de la serie que hayas visualizado: Psych
Por favor, introduce el grado de satisfaccion de la serie que hayas visualizado(1 -
-----
¡Serie modificada con éxito!
-----
¿Entrar en series?(si?)si
¿Quiere agregar serie visualizada o eliminar?agregar
Por favor, introduce el titulo de la serie que hayas visualizado: Breaking Bad
Por favor, introduce el grado de satisfaccion de la serie que hayas visualizado(1 -
-----
¡Serie modificada con éxito!
-----
¿Entrar en series?(si?)si
¿Quiere agregar serie visualizada o eliminar?agregar
Por favor, introduce el titulo de la serie que hayas visualizado: White Collar
Por favor, introduce el grado de satisfaccion de la serie que hayas visualizado(1 -

```

▼ Eliminar Usuario

Manual de uso: Eliminar Usuario

Para eliminar un usuario es necesario ejecutar la función `eliminarUsuario()`, esta función solicitará el DNI del usuario que se quiere eliminar y la borra del DataFrame `usuarioData`:

1. Se introduce el DNI del usuario que se quiere eliminar

Una vez eliminado el usuario el sistema devuelve la lista de usuarios almacenados.

```

def eliminarUsuario():
    DNI = input("Por favor, introduce el DNI del usuario que desea eliminar: ")
    global usuarioData
    global gustosData
    global series_visData

    usuarioData = usuarioData.drop(usuarioData[usuarioData['DNI']==DNI].index)
    gustosData = gustosData.drop(gustosData[gustosData['DNI']==DNI].index)
    series_visData = series_visData.drop(series_visData[series_visData['DNI']==DNI].index)

    usuarioData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/usuario.csv')
    gustosData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/Gustos_generales.csv')
    series_visData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/Series_visualizadas.csv')

    print("-----")
    print(" ¡Usuario eliminado con éxito!")
    print("-----")

```

```
return usuarioData
eliminarUsuario()
```

Por favor, introduce el DNI del usuario que desea eliminar: 45144479X

```
-----
¡Usuario eliminado con éxito!
-----
```

	DNI	Login	Password	Email	Telefono
0	24578169L	admin	admin	admin@gmail.com	758884194

▼ Sistema de recomendación basado en contenido

▼ Preparación recomendador

Aquí se muestran las ejecuciones necesarias para inicializar el recomendador

Manual de uso

1º

En este bloque con el uso de una bolsa de palabras cogeremos las palabras más relevantes de las descripciones de cada serie, y las mostramos en una nueva columna denominada "Descripcion_Procesada"

```
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer

import nltk
nltk.download('punkt')
nltk.download('stopwords')

ps = PorterStemmer()

descripcionProcesado = []

for column in seriesData.itertuples():

    descripcion = word_tokenize(str(column[4]))
    stops = set(stopwords.words("english"))
    descripcion = [ps.stem(w) for w in descripcion if not w in stops and w.isalnum()]
    descripcion = " ".join(descripcion)

    descripcionProcesado.append(descripcion)

preprocesadoData = seriesData
preprocesadoData['Descripcion_Procesada'] = descripcionProcesado
```


preprocesadoData

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
```

	Actores	Director	Genero	Descripcion	Anho	Titulo	Descripcion_Procesada
0	Rami Malek, Christian Slater, Carly Chaikin, M...	Sam Esmail	Crime, Drama, Thriller	Elliot, a brilliant but highly unstable young ...	24 Jun 2015	Mr. Robot	elliott brilliant h unstabl young engin
1	Cillian Murphy, Paul Anderson, Helen McCrory, ...	NaN	Crime, Drama	A gangster family epic set in 1919 Birmingham,...	30 Sep 2014	Peaky Blinders	A gangster famili epi 1919 birmingham e
2	Karoline Eichhorn, Louis Hofmann, Jödis Trieb...	NaN	Crime, Drama, Mystery, Sci-Fi, Thriller	A family saga with a supernatural twist, set i...	01 Dec 2017	Dark	A famili saga superr twist set german to
3	Bryan Cranston, Anna Gunn, Aaron Paul, Betsy B...	NaN	Crime, Drama, Thriller	A high school chemistry teacher diagnosed with...	20 Jan 2008	Breaking Bad	A high school chen teacher diagnos ino
	Peter Dinklage, Lena		Action, Adventure,	Nine noble families fight	17	Game of	nine nobl famili

2º

Con la métrica coseno calcularemos la distancia de cada serie entre ellas para saber lo que se asemejan unas a otras y poder hacer así la recomendación.

Se utiliza una distancia de vector coseno ya que tras probar el resto nos parece la que mejores resultados nos proporciona, junto con la euclídea. Su funcionamiento es el siguiente: Crea un espacio vectorial con n dimensiones correspondientes a los n items, situando los vectores según sus datos. Luego calcula el ángulo entre los vectores partiendo de la "coordenada cero". Menor distancia significa mayor similitud.

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import pairwise_distances

bagOfWordsModel = TfidfVectorizer()
bagOfWordsModel.fit(preprocesadoData[ 'Descripcion_Procesada' ])
```

```
textsBow= bagOfWordsModel.transform(preprocesadoData['Descripcion_Procesada'])
distanciaMatriz= pairwise_distances(textsBow,textsBow ,metric='cosine')
```

▼ Sistema de recomendación contenido serie

Manual de uso: Sistema de recomendación basado en contenido

Para recomendar una serie basandonos en su contenido será necesario ejecutar la función `recomendacionSerie()`, esta pedirá el nombre de la serie según la que queremos tener recomendaciones.

1. Se introduce el título de la serie

Se muestran las series recomendadas de esta.

```
def recomendacionSerie():
    tituloBuscado = input ("Por favor, introduce el título de la serie por la que quieres q
indexTitulo = preprocesadoData[preprocesadoData['Titulo']==tituloBuscado].index.values[0]
    titulosOrdenados = sorted(list(enumerate(distanciaMatriz[indexTitulo])), key=lambda x: >
    topTitulos = titulosOrdenados[1:11]
    topIndexes = [i[0] for i in topTitulos]
    return preprocesadoData['Titulo'].iloc[topIndexes]
```

```
recomendacionSerie()
```

```
Por favor, introduce el título de la serie por la que quieres que te recomendemos: D
33 Stranger Things
20 Big Love
62 Ray Donovan
32 Cable Girls
47 Genius
37 The Rain
77 Supernatural
22 Here and Now
78 Deadwood
8 The Simpsons
Name: Titulo, dtype: object
```

▼ Sistema de recomendación según series vistas

Manual de uso: Sistema de recomendación según series vistas

Para que un usuario obtenga las series que se le recomiendan será necesario ejecutar la función `recomendacionSerieSugerida()`, esta pedirá el DNI del usuario. Una vez se le pase un DNI comprobará las series que le gustan a ese usuario y de cada una de ellas cogerá las dos series que más se le asemejen a cada una y las mostrará.

1. Se introduce el DNI del usuario

Se muestran las series recomendadas para este usuario según las series que le han gustado.

```
def recomendacionSerieSugerida():
    usuarioBuscado = input ("Por favor, introduce el DNI del usuario que quiere ser recomend
    seriesVistas = series_visData[series_visData['DNI']==usuarioBuscado]
    numSeriesVistas = len(seriesVistas)
    i=0
    seriesVistasGustaArray = [0] #necesario crear un array
    seriesVistasGustaArray.clear() #borrar lo que haya antes de entrar al bucle
    while i < numSeriesVistas:
        if seriesVistas['Satisfaccion'][i]=="positive":
            seriesVistasGustaArray.append(seriesVistas['Titulo'][i])
            i+=1
    i=0
    seriesSugeridas = [0] #necesario crear un array
    seriesSugeridas.clear() #borrar lo que haya antes de entrar al bucle
    while i < len(seriesVistasGustaArray):
        indexTitulo = preprocesadoData[preprocesadoData['Titulo']==seriesVistasGustaArray[i]].
        titulosOrdenados = sorted(list(enumerate(distanciaMatriz[indexTitulo])), key=lambda x:
        topTitulos = titulosOrdenados[1:3]
        topIndexes = [j[0] for j in topTitulos]
        seriesSugeridas.append(preprocesadoData['Titulo'].iloc[topIndexes])
        i+=1
    return seriesSugeridas

recomendacionSerieSugerida()
```

```
Por favor, introduce el DNI del usuario que quiere ser recomendado: 41282090S
[33    Stranger Things
 20         Big Love
Name: Titulo, dtype: object, 83    Prodigal Son
103         Ballers
Name: Titulo, dtype: object]
```

▼ Sistema de análisis de sentimientos

▼ Preparación análisis de sentimientos

Aquí se muestran las ejecuciones necesarias para inicializar el sistema de analisis de sentimientos

Manual de uso

Primero elegiremos el modelo de clasificación con el que entrenaremos nuestro sistema en este caso escogemos Sigmoid, esto se debe a la mayor precisión en cuanto a valorar nuestros comentarios sobre series. Junto con linear han sido los que mayor porcentaje de acierto han tenido.

Utilizamos el procedimiento anterior de la bolsa de palabras para coger sólo las más relevantes y tras esto entrenamos nuestro sistema con el csv que tenemos en "analysisData".

```

from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn import svm
svc = svm.SVC(kernel='sigmoid') #Modelo de clasificación

import nltk
nltk.download('punkt')
nltk.download('stopwords')

ps = PorterStemmer()

preprocessedText = []

for row in analisisData.itertuples():

    text = word_tokenize(row[1]) ## indice de la columna que contiene el texto
    ## Remove stop words
    stops = set(stopwords.words("english"))
    text = [ps.stem(w) for w in text if not w in stops and w.isalnum()]
    text = " ".join(text)

    preprocessedText.append(text)

preprocessedData = analisisData
preprocessedData['processed_text'] = preprocessedText

bagOfWordsModel = TfidfVectorizer()
bagOfWordsModel.fit(preprocessedData['processed_text'])
textsBoW= bagOfWordsModel.transform(preprocessedData['processed_text'])

X_train = textsBoW #Documentos
Y_train = analisisData['sentiment'] #Etiquetas de los documentos
svc.fit(X_train, Y_train) #Entrenamiento

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='sigmoid',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)

```

▼ Analisis de sentimientos de comentario

Manual de uso: Analisis de sentimientos de comentario

Para el análisis de sentimientos se pedirá al usuario su DNI y la serie a valorar, tras esto le permitirá escribir un comentario y le dará la opción de poner una valoración en caso de no poner nada se realizará una valoración automática.

1. Introduce datos pedidos

Se muestra el comentario y la valoración (positive/negative) y se guarda en la base de datos.

```
def analisisSentimientos():
    usuarioComentario = input ("Por favor, introduce el DNI del usuario que quiere comentar
    serieComentario = input ("Por favor, introduce el titulo de la serie que se va a comentar
    comentarioComentario = input ("Por favor, introduce el comentario(english): ")
    valoracionComentario = input ("Por favor, introduce la valoración si lo desea(positive/negative): ")

    if valoracionComentario == "":
        ps = PorterStemmer()

        preprocessedText = []

        text = word_tokenize(comentarioComentario) ## indice de la columna que contiene el texto
        ## Remove stop words
        stops = set(stopwords.words("english"))
        text = [ps.stem(w) for w in text if not w in stops and w.isalnum()]
        text = " ".join(text)

        preprocessedText.append(text)

        textsBowTest= bagOfWordsModel.transform(preprocessedText)
        X_test = textsBowTest #Documentos
        valoracionComentario = svc.predict(X_test) #Se almacena en el array predictions las predicciones
        print(valoracionComentario)

    else:
        print(valoracionComentario)

    series_comentariosData.loc[series_comentariosData.shape[0]] = [usuarioComentario,serieComentario,comentarioComentario,valoracionComentario]
    series_comentariosData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/Series_visiData.csv')
    series_visData.loc[series_visData.shape[0]] = [usuarioComentario,serieComentario,comentarioComentario,valoracionComentario]
    series_visData.to_csv('/content/drive/My Drive/ABP_4_Teledocencia/SPRINT 2/CSV/Series_visiData.csv')
    print ("-----")
    print (" ¡Comentario añadida con éxito!")
    print ("-----")
```

analisisSentimientos()

```
Por favor, introduce el DNI del usuario que quiere comentar una serie: 68378360B
Por favor, introduce el titulo de la serie que se va a comentar: Dark
Por favor, introduce el comentario(english): good
Por favor, introduce la valoración si lo desea(positive/negative): positive
positive
-----
¡Comentario añadida con éxito!
-----
```

Bibliografía

-Apuntes de la asignatura

-Kaggle

-<https://www.aprendemachinellearning.com/>