Nombre completo: **José Gael Leyva Alvarado.**

Materia: **Introducción a Ciencia de Datos.**

Nombre del profesor: **Jaime Alejandro Romero Sierra.**

Fecha de entrega: **20-10-2025**

Repositorio de GitHub: https://github.com/ACAT12G/Proyecto-CienciadeDatos1/blob/main/README.md#proyecto-cienciadedatos1

Contexto de la base de datos:

Esta base de datos proviene de la plataforma de Kaggle, y es una simulación que representa diferentes interacciones que usuarios hipoteticos hayan tenido con un "*chat bot*", con diferentes resultados, a manera de obtener varias posibles situaciones listas para analisis.

Un "*chat bot*", es en español, un bot de chat, una inteligencia artificial especializada en hablar con los usuarios, da respuestas dependiendo en lo que el usuario quiera, y tiene el objetivo principal de asistir y/o automatizar ciertos tipos de trabajos de escritura.

En este caso especifico usa como usuarios a estudiantes, y por ende, los temas son desde estudio, hasta resumenes.

Los datos que se tienen de esta base de datos son:

Los datos que se tienen de esta base de datos son:

- ID de la sesión: El identificador de cada sesión que cuenta los estudiantes que tuvieron una sesión con el "*chat bot*".
- Nivel del estudiante: Preparatoria, Universidad y Graduado, determina el nivel del estudiante al momento de usar el "*chat bot*".
- Disciplina: Todas las materias en la que la conversación se formo: ingenieria, biologia, ciencias de la computación, matematicas, historia, negocios y psicologia.
- Fecha de la sesión: La fecha en la que la conversación se llevo acabo.
- Mensajes totales: El total de mensajes que tuvo el usuario y la Inteligencia Artificial; la extensión de la conversación.
- Tipo de Tarea: Escritura, estudiar, codificar, asistencia con la tarea, lluvia de ideas e investigación; todas las actividades que los estudiantes llevaron acabo con el "*chat bot*".
- Nivel de Asistencia de la Inteligencia Artificial: En un rango de 1-5, determina si la IA fue de ayuda y asisistio de manera idonea.
- Resultado Final: Nos dice si la tarea fue realizada como se debe, si fue aplazada, si el estudiante termino con dudas/confundid@, o si el estudiante se rindio.
- ¿Se uso de nuevo?: Determina si el estudiante volvio a usar el "*chat bot*", o no.
- Indice de Satisfacciom: Calificación final que el estudiante otorgo a su experiencia con el "*chat bot*".

Limpieza de base de datos:

Exploracion:

La exploración se basa en ver que problemas tenemos, que buscamos solucionar y en general, el parnorama general de a que nos enfrentamos.

```
[95]  ✓ 0.0s                                                    Python
```

```python
import pandas as pd

df = pd.read_csv("df_sucio.csv")

df.isnull().sum()
```

```
SessionID            540
StudentLevel         540
Discipline           540
SessionDate          540
SessionLengthMin     540
TotalPrompts         540
TaskType             540
AI_AssistanceLevel   540
FinalOutcome         540
UsedAgain            540
SatisfactionRating   540
dtype: int64
```

Nos muestra todos los valores NaN, o nulos que tiene cada fila.

```
[96]  df["SessionID"].unique()
      ✓ 0.0s                                                    Python
...   array(['SESSION00001', 'SESSION00002', nan, ..., 'SESSION00102',
             'SESSION08149', 'SESSION02496'], shape=(9366,), dtype=object)
```

Nuestra base de datos en estado inicial, se pueden ver los NaN desde aqui.

```
[97]  df
      ✓ 0.0s                                                    Python
```

	SessionID	StudentLevel	Discipline	SessionDate	SessionLengthMin	TotalPrompts	TaskType	AI_AssistanceLevel	FinalOutcome	UsedAgain	S
0	SESSION00001	Undergraduate	Computer Science	2024-11-03	31.20	11	Studying	2.0	Assignment Completed	True	
1	SESSION00002	Undergraduate	NaN	2024-08-25	13.09	6	Studying	3.0	Assignment Completed	True	
2	NaN	Undergraduate	Business	2025-01-12	19.22	5	Coding	3.0	Assignment Completed	True	
3	SESSION00004	Undergraduate	Computer Science	2025-05-06	3.70	1	Coding	3.0	NaN	True	
4	SESSION00005	Undergraduate	Psychology	2025-03-18	28.12	9	Writing	3.0	Assignment Completed	True	
...	...	...	...	...	...	...	...	...	...	...	
10803	SESSION03028	Graduate	Math	2025-03-04	20.52	NaN	Coding	4.0	Assignment Completed	NaN	
10804	SESSION01621	High School	Computer Science	2024-12-12	40.27	9	Research	4.0	Gave Up	False	
10805	SESSION07276	High School	History	2025-05-02	41.14	9	Writing	4.0	NaN	False	
10806	SESSION06220	Graduate	Business	2024-11-15	NaN	12	Research	3.0	Assignment Completed	True	
10807	SESSION01851	Undergraduate	Biology	2025-01-29	28.86	NaN	Coding	5.0	Confused	True	

```
[100] df.info()
      ✓ 0.0s
...   <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 10808 entries, 0 to 10807
      Data columns (total 11 columns):
       #   Column             Non-Null Count  Dtype
      ---  ------             --------------  -----
       0   SessionID          10268 non-null  object
       1   StudentLevel       10268 non-null  object
       2   Discipline         10268 non-null  object
       3   SessionDate        10268 non-null  object
       4   SessionLengthMin   10268 non-null  float64
       5   TotalPrompts       10268 non-null  object
       6   TaskType           10268 non-null  object
       7   AI_AssistanceLevel 10268 non-null  float64
       8   FinalOutcome       10268 non-null  object
       9   UsedAgain          10268 non-null  object
       10  SatisfactionRating 10268 non-null  float64
      dtypes: float64(3), object(8)
      memory usage: 928.9+ KB
```

Nos muestra los tipos de datos de cada columna.

Traducción:

El primer paso es traducir todo a español, para evitar futuros conflictos, mas acerca de que tipo de problemas en el final de este documento.

```
dic_renombrar = {
    'SessionID': 'ID_Sesion',
    'StudentLevel': 'NivelEstudiante',
    'Discipline': 'Disciplina',
    'SessionDate': 'FechaSesion',
    'SessionLengthMin': 'DuracionMinutos',
    'TotalPrompts': 'TotalIndicaciones',
    'TaskType': 'TipoTarea',
    'AI_AssistanceLevel': 'NivelAsistenciaIA',
    'FinalOutcome': 'ResultadoFinal',
    'UsedAgain': 'UsadoNuevamente',
    'SatisfactionRating': 'CalificacionSatisfaccion'
}

traduccion_nivel = {
    'Undergraduate': 'Universitario',
    'High School': 'Preparatoria',
    'Graduate': 'Graduado'
}

traduccion_disciplina = {
    'Engineering': 'Ingeniería',
    'Computer Science': 'Ciencias de la Computación',
    'History': 'Historia',
    'Biology': 'Biología',
    'Math': 'Matemáticas',
    'Business': 'Negocios',
    'Psychology': 'Psicología'
}

traduccion_tarea = {
    'Writing': 'Redacción',
    'Studying': 'Estudio',
    'Coding': 'Programación',
    'Homework Help': 'Ayuda con Tareas',
    'Brainstorming': 'Lluvia de Ideas',
    'Research': 'Investigación'
}

traduccion_resultado = {
    'Assignment Completed': 'Tarea Completada',
    'Idea Drafted': 'Idea Rezagada',
    'Confused': 'Confundido'
}

df2.rename(columns=dic_renombrar, inplace=True)

traducciones_completas = {
    'NivelEstudiante': traduccion_nivel,
    'Disciplina': traduccion_disciplina,
    'TipoTarea': traduccion_tarea,
    'ResultadoFinal': traduccion_resultado
}
```

Nos muestra los tipos de datos de cada columna.

```
[101]  df.columns
       ✓ 0.0s

...    Index(['SessionID', 'StudentLevel', 'Discipline', 'SessionDate',
              'SessionLengthMin', 'TotalPrompts', 'TaskType', 'AI_AssistanceLevel',
              'FinalOutcome', 'UsedAgain', 'SatisfactionRating'],
             dtype='object')
```

Depuracion de NaN:

Utilizando fillna, se usa un codigo para distruiir todos los NaN rellnados con valores aleatorios dentro del rango de datos que tenemos, a manera de no favorecer ningun tipo de dato.

```
[110]  df2["ID_Sesion"].unique()
       ✓ 0.0s                                                    Python

...    array(['SESSION00001', 'SESSION00002', nan, ..., 'SESSION00102',
              'SESSION08149', 'SESSION02496'], shape=(9366,), dtype=object)
```

```
[111]  df2['ID_Sesion'] = df2['ID_Sesion'].fillna("SESSION00000")
       df2.isnull().sum()
       ✓ 0.0s                                                    Python

...    ID_Sesion                  0
       NivelEstudiante          540
       Disciplina               540
       FechaSesion              540
       DuracionMinutos          540
       TotalIndicaciones        540
       TipoTarea                540
       NivelAsistenciaIA        540
       ResultadoFinal           540
       UsadoNuevamente          540
       CalificacionSatisfaccion 540
       dtype: int64
```

Código especial:

```python
import numpy as np

valores_validos = df['NivelEstudiante'].dropna()

Identifica los índices en df2 que tienen NaN
indices_nan = df2.loc[df2['NivelEstudiante'].isnull()].index

Genera el número exacto de valores aleatorios que necesitas
valores_aleatorios = np.random.choice(valores_validos, size=len(indices_nan))

Asigna estos valores aleatorios directamente a las ubicaciones NaN usando .loc
df2.loc[indices_nan, 'NivelEstudiante'] = valores_aleatorios

df2.isnull().sum()
```

```
valores_validos = df['ResultadoFinal'].dropna()

indices_nan = df2.loc[df2['ResultadoFinal'].isnull()].index

valores_aleatorios = np.random.choice(valores_validos, size=len(indices_nan))

df2.loc[indices_nan, 'ResultadoFinal'] = valores_aleatorios

df2.isnull().sum()
```

[134] ✓ 0.0s

```
ID_Sesion                0
NivelEstudiante          0
Disciplina               0
FechaSesion              0
DuracionMinutos          0
TotalIndicaciones        0
TipoTarea                0
NivelAsistenciaIA        0
ResultadoFinal           0
UsadoNuevamente          0
CalificacionSatisfaccion 0
dtype: int64
```

Depuracion de duplicados:

```
df2=df2.drop_duplicates(subset=['ID_Sesion'])
df2
```

[39] ✓ 0.0s Python

	ID_Sesion	NivelEstudiante	Disciplina	FechaSesion	DuracionMinutos	TotalIndicaciones	TipoTarea	NivelAsistenciaIA	ResultadoFinal	Usad
0	SESSION00001	Universitario	Ciencias de la Computación	2024-11-03	31.20	11	Estudio	2.0	Tarea Completada	
1	SESSION00002	Universitario	Negocios	2024-08-25	13.09	6	Estudio	3.0	Tarea Completada	
2	SESSION00000	Universitario	Negocios	2025-01-12	19.22	5	Programación	3.0	Tarea Completada	
3	SESSION00004	Universitario	Ciencias de la Computación	2025-05-06	3.70	1	Programación	3.0	Idea Rezagada	
4	SESSION00005	Universitario	Psicología	2025-03-18	28.12	9	Redacción	3.0	Tarea Completada	
...	...	...	...	...	...	...	...	...	...	
10729	SESSION07940	Universitario	Psicología	2025-02-09	56.13	24	Investigación	2.0	Confundido	
10741	SESSION04427	Preparatoria	Ingeniería	2024-08-28	16.97	3	Estudio	4.0	Gave Up	
10758	SESSION00102	Universitario	Ingeniería	2025-01-18	16.16	4	Estudio	4.0	Tarea Completada	
10770	SESSION08149	Graduado	Ingeniería	2024-12-22	20.55	5	Redacción	4.0	Tarea Completada	
10786	SESSION02496	Universitario	Matemáticas	2025-05-14	5.85	1	Lluvia de Ideas	5.0	Idea Rezagada	

```
[40]  df2.shape
      ✓ 0.0s

...   (9366, 11)
```

```
[41]  df2.duplicated().sum()
      ✓ 0.0s

...   np.int64(0)
```

Buscar y eliminar si lo hay, texto especifico:

El "bbb" es nuestro texto expecifico, hay que eliminarlo. Pero primero, hay que buscarlo.

```
[136]  df2['ID_Sesion'].unique()
       ✓ 0.0s

...    array(['SESSION00001', 'SESSION00002', 'SESSION00000', ...,
              'SESSION00102', 'SESSION08149', 'SESSION02496'],
             shape=(9366,), dtype=object)
```

```
[137]  df2['NivelEstudiante'].unique()
       ✓ 0.0s

...    array(['Universitario', 'Graduado', 'Preparatoria'], dtype=object)
```

```
[138]  df2['Disciplina'].unique()
       ✓ 0.0s

...    array(['Ciencias de la Computación', 'Negocios', 'Psicología', 'Biología',
              'Matemáticas', 'Historia', 'Ingeniería', 'bbb'], dtype=object)
```

```
[140]  df2['FechaSesion'].unique()
       ✓ 0.0s

...    array(['2024-11-03', '2024-08-25', '2025-01-12', '2025-05-06',
```

Detectando valores especificos:

Una vez encontramos aquellos que tienen el texto especifico, buscamos cuanto de este texto hay.

```
df2[df2['Disciplina'] == 'bbb'].shape[0]
```
✓ 0.0s

```
215
```

```
df2[df2['UsadoNuevamente'] == 'bbb'].shape[0]
```
✓ 0.0s

```
216
```

```
df2[df2['TipoTarea'] == 'bbb'].shape[0]
```

```
208
```

```
df2[df2['TotalIndicaciones'] == 'bbb'].shape[0]
```
✓ 0.0s

```
209
```

Eliminando valores especificos:

```
df2=df2[df2['Disciplina'] != 'bbb']
```
[152] ✓ 0.0s

```
df2=df2[df2['UsadoNuevamente'] != 'bbb']
```
[153] ✓ 0.0s

```
df2=df2[df2['TipoTarea'] != 'bbb']
```
[154] ✓ 0.0s

```
df2=df2[df2['TotalIndicaciones'] != 'bbb']
```
[155] ✓ 0.0s

Comprobación:

```
df2[df2['Disciplina'] == 'bbb'].shape[0]
```
[156] ✓ 0.0s

```
0
```

```
df2[df2['UsadoNuevamente'] == 'bbb'].shape[0]
```
[157] ✓ 0.0s

```
0
```

```
df2[df2['TipoTarea'] == 'bbb'].shape[0]
```
[158] ✓ 0.0s

```
0
```

```
df2[df2['TotalIndicaciones'] == 'bbb'].shape[0]
```
[159] ✓ 0.0s

```
0
```

```
df2.shape
```
[160] ✓ 0.0s

```
(9983, 11)
```

Cambio del tipo de variables:

Algunas columnas no son del tipo que deben ser, hay que cambiar eso,

```
df2['TotalIndicaciones']=df2['TotalIndicaciones'].astype(float)
```
[163] ✓ 0.0s

```
df2['TotalIndicaciones']=df2['TotalIndicaciones'].astype(int)
```
[164] ✓ 0.0s

```
df2['NivelAsistenciaIA']=df2['NivelAsistenciaIA'].astype(int)
```
[165] ✓ 0.0s

```
df2['UsadoNuevamente']=df2['UsadoNuevamente'].astype(bool)
```
[166] ✓ 0.0s

```
df2.info()
```

[167] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
Index: 9983 entries, 0 to 10807
Data columns (total 11 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   ID_Sesion               9983 non-null   object
 1   NivelEstudiante         9983 non-null   object
 2   Disciplina              9983 non-null   object
 3   FechaSesion             9983 non-null   object
 4   DuracionMinutos         9983 non-null   float64
 5   TotalIndicaciones       9983 non-null   int64
 6   TipoTarea               9983 non-null   object
 7   NivelAsistenciaIA       9983 non-null   int64
 8   ResultadoFinal          9983 non-null   object
 9   UsadoNuevamente         9983 non-null   bool
 10  CalificacionSatisfaccion  9983 non-null   float64
dtypes: bool(1), float64(2), int64(2), object(6)
memory usage: 867.7+ KB
```

Resultado Final:

```
df2
```

[172] ✓ 0.0s Python

	ID_Sesion	NivelEstudiante	Disciplina	FechaSesion	DuracionMinutos	TotalIndicaciones	TipoTarea	NivelAsistenciaIA	ResultadoFinal	Usad
0	SESSION00001	Universitario	Ciencias de la Computación	2024-11-03	31.20	11	Estudio	2	Tarea Completada	
1	SESSION00002	Universitario	Ciencias de la Computación	2024-08-25	13.09	6	Estudio	3	Tarea Completada	
2	SESSION00000	Universitario	Negocios	2025-01-12	19.22	5	Programación	3	Tarea Completada	
3	SESSION00004	Universitario	Ciencias de la Computación	2025-05-06	3.70	1	Programación	3	Tarea Completada	
4	SESSION00005	Universitario	Psicología	2025-03-18	28.12	9	Redacción	3	Tarea Completada	
...	...	...	...	...	...	...	...	...	...	
10803	SESSION03028	Graduado	Matemáticas	2025-03-04	20.52	7	Programación	4	Tarea Completada	
10804	SESSION01621	Preparatoria	Ciencias de la Computación	2024-12-12	40.27	9	Investigación	4	Gave Up	
10805	SESSION07276	Preparatoria	Historia	2025-05-02	41.14	9	Redacción	4	Gave Up	
10806	SESSION06220	Graduado	Negocios	2024-11-15	37.07	12	Investigación	3	Tarea Completada	

```
df2.isnull().sum()
```

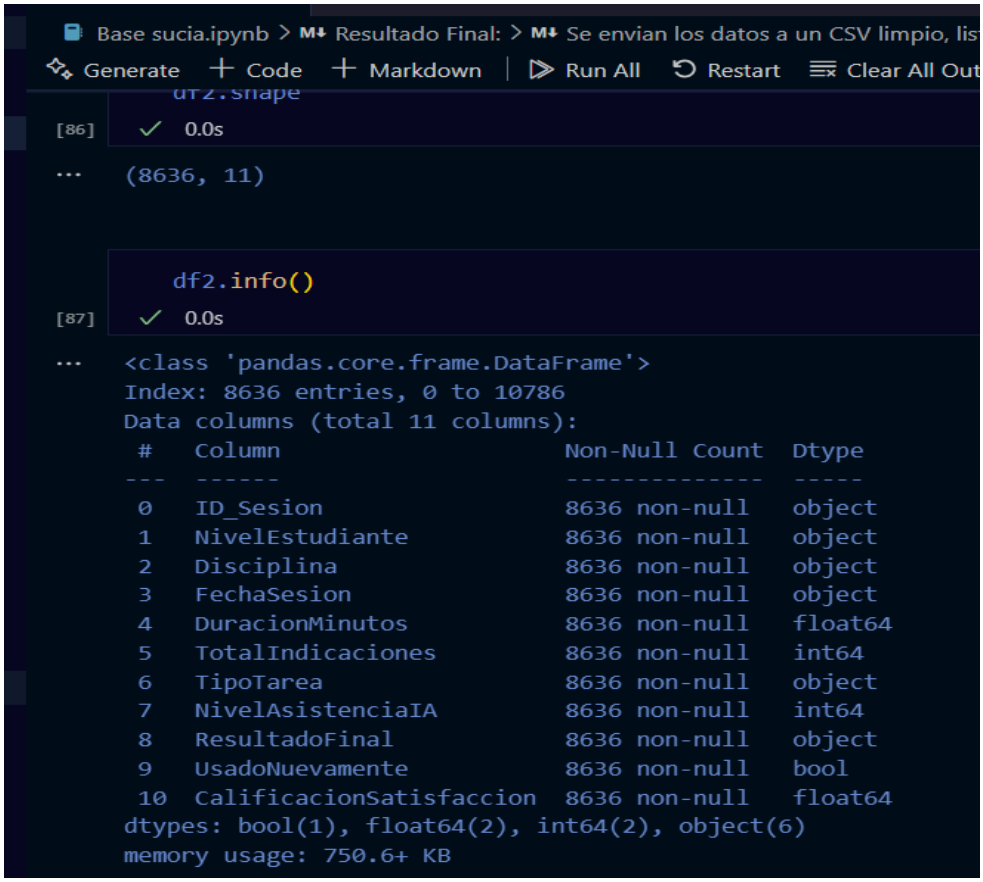
[173] ✓ 0.0s

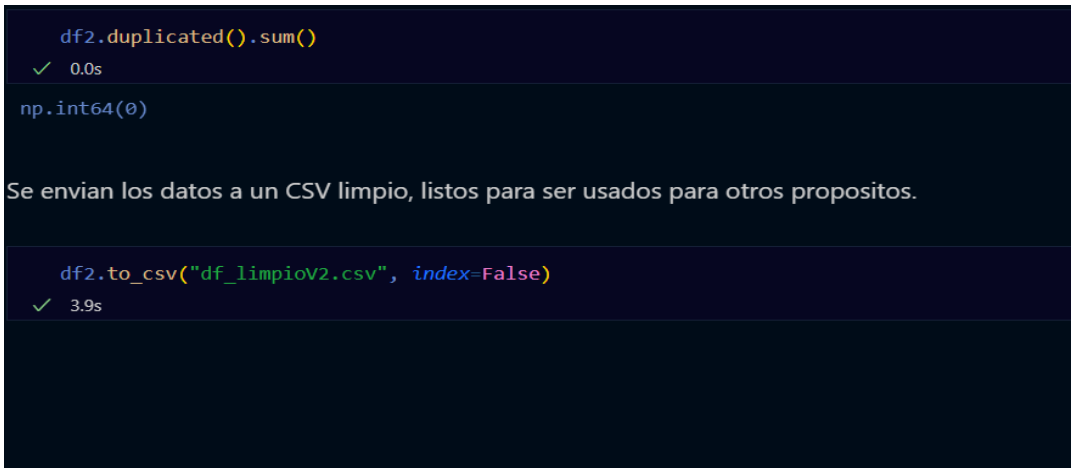
```
ID_Sesion              0
NivelEstudiante        0
Disciplina             0
FechaSesion            0
DuracionMinutos        0
TotalIndicaciones      0
TipoTarea              0
NivelAsistenciaIA      0
ResultadoFinal         0
UsadoNuevamente        0
CalificacionSatisfaccion    0
dtype: int64
```

```
df2.shape
```

[174] ✓ 0.0s

```
(9983, 11)
```





Problemas Principales:

Hubo un problema con la traducción, pues esta tenía la tendencia a alterar el resultado final, y revertir mucho del trabajo efectuado.

Soluciones:

Para la traducción, solo es realizarla desde un inicio, eso evita todo el conflicto y/o perdida de tiempo.

Aprendizajes:

Mis aprendizajes finales fue que, con una organización y empezar con fundamentos ofrece más velocidad y eficiencia con el procedimiento.