


PRACTICA 2

De laboratorio 'Programación en python'

ALEXA ESMERALDA SALDA  A MORALES
UNIDAD2

Práctica de Laboratorio - Unidad 2: Introducción a la Limpieza de

Datos Materia: Introducción a la Ciencia de Datos

Unidad 2: Procesamiento y Limpieza de Datos

Título de la Práctica: Limpieza de una Base de Datos Ensuciada

Nombre: Saldaña Morales Alexa Esmeralda

Día y horario de la materia: lunes 9:00 a 10:00
martes y jueves 8:00 a 10:00

Objetivo:

Desarrollar habilidades en el preprocesamiento de datos, incluyendo la identificación y tratamiento de valores faltantes, datos duplicados, y formatos inconsistentes en una base de datos.

Instrucciones:

1. Recepción de la Base de Datos

Descargue el archivo de la base de datos "ensuciada" proporcionado por el profesor. Este archivo ha sido alterado con errores comunes, como valores duplicados, valores faltantes (NaNs), y errores en el formato de algunas columnas.

2. Análisis Inicial de la Base de Datos

Antes de comenzar a limpiar la base de datos, debe realizar un análisis preliminar para comprender la naturaleza y distribución de los errores.

- Visualización de la base de datos inicial.
Podemos notar que tenemos un total de 6158 datos, pero existen columnas que mas adelante no serán tan necesarias.

+ Código + Texto

```
import pandas as pd
import numpy as np
url="https://raw.githubusercontent.com/Alexasald/df-sucio-final/refs/heads/main/df-sucio.csv"
df= pd.read_csv(url)
df
```

	Employee_ID	Age	Gender	Job_Role	Industry	Years_of_Experience	Work_Location	Hours_Worked_Per_Week	Number_of_Virtual_Meetings	Work-Life_Balance_Rating	Stress_Level	Mental_Health_Condition	Access_to_Mental_He
0	EMP0001	32	Non-binary	HR	Healthcare	13.0	Hybrid	47.0	7.0	2.0	Medium	Depression	
1	EMP0002	40	Female	Data Scientist	IT	3.0	Remote	52.0	4.0	1.0	Medium	Anxiety	
2	EMP0003	59	Non-binary	Software Engineer	Education	22.0	Hybrid	46.0	11.0	5.0	Medium	Anxiety	
3	EMP0004	27	Male	Software Engineer	Finance	20.0	Onsite	32.0	8.0	4.0	High	Depression	
4	EMP0005	49	Male	Sales	Consulting	32.0	Onsite	35.0	12.0	2.0	High	NaN	
...
6153	EMP2944	58	Male	Data Scientist	Education	19.0	Remote	34.0	5.0	2.0	Medium	NaN	
6154	EMP2019	38	Prefer not to say	Software Engineer	Manufacturing	27.0	Remote	49.0	13.0	3.0	High	Depression	
6155	EMP3339	59	Non-binary	Marketing	Manufacturing	20.0	Onsite	21.0	11.0	3.0	High	Burnout	
6156	EMP3228	44	Prefer not to say	Marketing	Education	NaN	bbb	54.0	8.0	4.0	Medium	NaN	
6157	EMP3818	53	Male	Marketing	IT	15.0	Remote	59.0	NaN	2.0	Medium	Anxiety	

6158 rows x 14 columns

Próximos pasos:

Generar código con AI

Ver gráficos recomendados

New interactive sheet

- **Resumen estadístico de los datos.**

con el comando de describe(), podemos visualizar la tabla con el conteo completo en casi todas las columnas excepto en la de actividad física. También se muestran la falta de columnas ya que el tipo de dato no concuerda con ser dato numérico por ello más adelante tendremos que modificar estos datos.

```
[450] #resumen estadístico de los datos
df.describe()
```

	Años_Expe	Hrs_semana	N_reun_V	Balance_vida_trabajo	Calificación_aislamientoS	Actividad_fisica
count	5851.000000	5851.000000	5851.000000	5851.000000	5851.000000	0.0
mean	17.900530	39.663818	7.558879	2.988378	3.002905	NaN
std	10.050676	11.872569	4.643983	1.413803	1.398103	NaN
min	1.000000	20.000000	0.000000	1.000000	1.000000	NaN
25%	9.000000	29.000000	4.000000	2.000000	2.000000	NaN
50%	18.000000	40.000000	8.000000	3.000000	3.000000	NaN
75%	26.000000	50.000000	12.000000	4.000000	4.000000	NaN
max	35.000000	60.000000	15.000000	5.000000	5.000000	NaN

- **Porcentaje de valores faltantes por columna**

El siguiente comando. isnull().mean(), muestra la tabla de porcentajes de cada columna, aquí podemos interpretar que en su mayoría los porcentajes de datos faltantes son del 4.9854% , la de salud mental es de 27.72% y la de actividad física de un 100% lo cual es preocupante ya que sus datos son nulos.

```
#porcentaje de valores faltantes inicial
df.isnull().mean()
```

	0
ID_E	0.049854
Edad	0.049854
Genero	0.049854
Rol	0.049854
Industria	0.049854
Años_Expe	0.049854
Ubicacion	0.049854
Hrs_semana	0.049854
N_reun_V	0.049854
Balance_vida_trabajo	0.049854
Nivel_estres	0.049854
Salud_Mental	0.277200
Recurso_SaludM	0.049854
Cambio_en_productividad	0.049854
Calificación_aislamientoS	0.049854
Satisfacción_trabajo	0.049854
Apoyo_de_empresa	0.049854
Actividad_fisica	1.000000
Calidad_del_sueño	0.049854
Region	0.049854

dtype: float64

- **Existencia duplicados**

Con el comando. `df.duplicated().sum` Se muestra el conteo total de filas duplicadas

```

[452] #Contar filas duplicadas
df.duplicated().sum()

145

```

- **Tipo de dato**

En la siguiente tabla observamos lo mencionado anteriormente, los datos están definidos de manera incorrecta lo cual nos provocara algunos conflictos para el análisis.

```

[333] #tipo de datos
df1.info()

<class 'pandas.core.frame.DataFrame'>
Index: 6013 entries, 0 to 6157
Data columns (total 20 columns):
#   Column              Non-Null Count  Dtype
---  -
0   ID_E                5706 non-null   object
1   Edad                5707 non-null   object
2   Genero              5708 non-null   object
3   Rol                 5706 non-null   object
4   Industria           5706 non-null   object
5   Años_Expe           5706 non-null   float64
6   Ubicacion           5707 non-null   object
7   Hrs_semana          5706 non-null   float64
8   N_reun_V            5707 non-null   float64
9   Balance_vida_trabajo 5707 non-null   float64
10  Nivel_estres         5706 non-null   object
11  Salud_Mental         4340 non-null   object
12  Recurso_SaludM       5708 non-null   object
13  Cambio_en_productividad 5706 non-null   object
14  Calificación_aislamientos 5706 non-null   float64
15  Satisfacción_trabajo  5707 non-null   object
16  Apoyo_de_empresa     5707 non-null   object
17  Actividad_fisica     0 non-null      float64
18  Calidad_del_sueño    5707 non-null   object
19  Region              5706 non-null   object
dtypes: float64(6), object(14)
memory usage: 986.5+ KB

```

Ahora al conocer un poco mas los datos con los que se trabajara y los detalles que se tendrán que tomar en cuenta, iniciare con el proceso de limpieza teniendo en cuenta el nuevo etiquetado de datos y la resolución de los datos faltantes.

Uno de los problemas es que la cantidad de datos faltantes es importante ya que en la siguiente fase de limpieza suelen perderse muchos de los datos, haciendo así la base de datos muy pequeña.

Limpieza de datos.

- **Eliminación de duplicados**

Agregue un nuevo DF1 para no trabajar sobre el mismo, quite los duplicados de todo el DF, anteriormente teníamos un total de 6158 y ahora eliminando duplicados nos quedo un total de 6013 datos

```
df1=df.drop_duplicates()  
df1
```

	ID_E	Edad	Genero	Rol	Industria	Años_Expe	Ubicacion	Hrs_semana	N_reun_V	Balance_vida_trabajo	Nivel_estres	Salud_Mental	Recurso_SaludM	Cambio_en_productividad	Calif
0	EMP0001	32	Non-binary	HR	Healthcare	13.0	Hybrid	47.0	7.0		2.0	Medium	Depression	NaN	bbb
1	EMP0002	40	Female	Data Scientist	IT	3.0	Remote	52.0	4.0		1.0	Medium	Anxiety	NaN	Increase
2	EMP0003	59	Non-binary	Software Engineer	Education	22.0	Hybrid	46.0	11.0		5.0	Medium	Anxiety	No	No Change
3	EMP0004	27	Male	Software Engineer	Finance	20.0	Onsite	32.0	8.0		4.0	High	Depression	Yes	Increase
4	EMP0005	49	Male	Sales	Consulting	32.0	Onsite	35.0	12.0		2.0	High	NaN	Yes	NaN
...
6152	EMP1138	28	Female	Marketing	NaN	5.0	Onsite	52.0	0.0		4.0	Low	Burnout	Yes	Increase
6153	EMP2944	58	Male	Data Scientist	Education	19.0	Remote	34.0	5.0		2.0	Medium	NaN	No	NaN
6155	EMP3339	59	Non-binary	Marketing	Manufacturing	20.0	Onsite	21.0	11.0		3.0	High	Burnout	No	Increase
6156	EMP3228	44	Prefer not to say	Marketing	Education	NaN	bbb	54.0	8.0		4.0	Medium	NaN	Yes	Decrease
6157	EMP3516	33	Male	Marketing	IT	15.0	Remote	59.0	NaN		2.0	Medium	Anxiety	bbb	No Change

6013 rows × 16 columns

- **Corrección de datos 'bbb'**

Coloque un ciclo para saber cuántos datos había escritos de esa manera, después realice otro ciclo for para eliminar aquellos datos inválidos. Dejando así solo datos NaN y los datos normales para después rellenar los datos vacíos.

Nota (profesor intentó convertirlos a datos NaN pero me las cadenas de caracteres no me dejaba colocarlos a datos nulos 😞)

```
[336] #numero de veces que esta 'bbb'
for i in lista:
    print(f'En la columna {i} la cadena de 'bbb' son: {df2[df2[i] == 'bbb'].shape[0]}')
```

```
En la columna ID_E la cadena de 'bbb' son: 0
En la columna Edad la cadena de 'bbb' son: 118
En la columna Genero la cadena de 'bbb' son: 0
En la columna Rol la cadena de 'bbb' son: 115
En la columna Industria la cadena de 'bbb' son: 0
En la columna Años_Expe la cadena de 'bbb' son: 0
En la columna Ubicacion la cadena de 'bbb' son: 115
En la columna Hrs_semana la cadena de 'bbb' son: 0
En la columna N_reun_V la cadena de 'bbb' son: 0
En la columna Balance_vida_trabajo la cadena de 'bbb' son: 0
En la columna Nivel_estres la cadena de 'bbb' son: 117
En la columna Salud_Mental la cadena de 'bbb' son: 115
En la columna Recurso_SaludM la cadena de 'bbb' son: 118
En la columna Cambio_en_productividad la cadena de 'bbb' son: 110
En la columna Calificación_aislamientoS la cadena de 'bbb' son: 0
En la columna Satisfacción_trabajo la cadena de 'bbb' son: 0
En la columna Apoyo_de_empresa la cadena de 'bbb' son: 111
En la columna Actividad_fisica la cadena de 'bbb' son: 0
En la columna Calidad_del_sueño la cadena de 'bbb' son: 114
En la columna Region la cadena de 'bbb' son: 0
```

```
#ciclo para eliminar 'bbb'
for i in lista:
    df2=df2[df2[i] != 'bbb']
df2
```

	ID_E	Edad	Genero	Rol	Industria	Años_Expe	Ubicacion	Hrs_semana	N_reun_V	Balance_vida_trabajo	Nivel_estres	Salud_Mental	Recurso_SaludM	Can
1	EMP0002	40	Female	Data Scientist	IT	3.0	Remote	52.0	4.0		1.0	Medium	Anxiety	NaN
2	EMP0003	59	Non-binary	Software Engineer	Education	22.0	Hybrid	46.0	11.0		5.0	Medium	Anxiety	No
3	EMP0004	27	Male	Software Engineer	Finance	20.0	Onsite	32.0	8.0		4.0	High	Depression	Yes
4	EMP0005	49	Male	Sales	Consulting	32.0	Onsite	35.0	12.0		2.0	High	NaN	Yes
5	EMP0006	59	Non-binary	Sales	IT	31.0	Hybrid	39.0	3.0		4.0	High	NaN	No
...
6149	EMP4544	56	Prefer not to say	Software Engineer	Education	5.0	Hybrid	35.0	3.0		5.0	Medium	Anxiety	Yes
6151	EMP3533	41	Prefer not to say	Project Manager	IT	9.0	Onsite	50.0	8.0		2.0	High	NaN	Yes
6152	EMP1138	28	Female	Marketing	NaN	5.0	Onsite	52.0	0.0		4.0	Low	Burnout	Yes
6153	EMP2944	58	Male	Data Scientist	Education	19.0	Remote	34.0	5.0		2.0	Medium	NaN	No
6155	EMP3339	59	Non-binary	Marketing	Manufacturing	20.0	Onsite	21.0	11.0		3.0	High	Burnout	No

5040 rows x 15 columns

• Corrección de tipos de datos:

Convertir solamente a datos numéricos aquellos que estaban etiquetados como object para poder usarlos como números y se reconocieran de esa manera.

```
[460] #cambio de tipo de dato a numerico
df3['Edad']=df3['Edad'].astype(float)
df3['Apoyo_de_empresa']=df3['Apoyo_de_empresa'].astype(float)
```

```
<ipython-input-460-25b2526f83fb>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df3['Edad']=df3['Edad'].astype(float)
<ipython-input-460-25b2526f83fb>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df3['Apoyo_de_empresa']=df3['Apoyo_de_empresa'].astype(float)
```

```
df3['Edad']=df3['Edad'].astype(int)
df3['Años_Expe']=df3['Años_Expe'].astype(int)
df3['Hrs_semana']=df3['Hrs_semana'].astype(int)
df3['Apoyo_de_empresa']=df3['Apoyo_de_empresa'].astype(int)
df3['Balance_vida_trabajo']=df3['Balance_vida_trabajo'].astype(int)
```

Como lo vimos anteriormente, la fila de actividad física tenía el 100% de datos faltantes por lo que decidí eliminarla, junto con el ID del empleado lo cual no se hace relevante en el análisis futuro

```
#elimino la columna de 'actividad_fisica' porque esta completaamnete vacia.
df3= df2.drop(columns=['ID_E','Actividad_fisica','N_reun_V',])
df3
```

✓ 0.0s

	Edad	Genero	Rol	Industria	Años_Expe	Ubicacion	Hrs_semana	Balance_vida_trabajo	Nivel_estres	Salud_Mental	Recurso_SaludM	Cambio_en_productividad	Califi
1	40.0	Female	Data Scientist	IT	3.0	Remote	52.0		1.0	Medium	Anxiety	NaN	Increase
2	59.0	Non-binary	Software Engineer	Education	22.0	Hybrid	46.0		5.0	Medium	Anxiety	No	No Change
3	27.0	Male	Software Engineer	Finance	20.0	Onsite	32.0		4.0	High	Depression	Yes	Increase
4	49.0	Male	Sales	Consulting	32.0	Onsite	35.0		2.0	High	NaN	Yes	NaN
5	59.0	Non-binary	Sales	IT	31.0	Hybrid	39.0		4.0	High	NaN	No	Increase
...
6149	56.0	Prefer not to say	Software Engineer	Education	5.0	Hybrid	35.0		5.0	Medium	Anxiety	Yes	Increase
6151	41.0	Prefer not to say	Project Manager	IT	9.0	Onsite	50.0		2.0	High	NaN	Yes	No Change
6152	28.0	Female	Marketing	NaN	5.0	Onsite	52.0		4.0	Low	Burnout	Yes	Increase
6153	58.0	Male	Data Scientist	Education	19.0	Remote	34.0		2.0	Medium	NaN	No	NaN
6155	59.0	Non-binary	Marketing	Manufacturing	20.0	Onsite	21.0		3.0	High	Burnout	No	Increase

5040 rows x 17 columns

- **Eliminación o imputación de valores faltantes:**

Rellene los valores nulos por moda y promedio, así no serían muchos datos eliminados o perdidos. Ya que como se vio en la tabla de porcentajes, era una cantidad considerable de datos faltantes lo que haría que el análisis fuera complicado y no muy preciso.

```
#rellenar datos faltan

df3['Edad']=df3['Edad'].fillna(df3['Edad'].mean())
df3['Genero']=df3['Genero'].fillna(df3['Genero'].mode()[0])
df3['Nivel_estres']=df3['Nivel_estres'].fillna(df3['Nivel_estres'].mode()[0])
df3['Rol']=df3['Rol'].fillna(df3['Rol'].mode()[0])
df3['Años_Expe']=df3['Años_Expe'].fillna(df3['Años_Expe'].mean())
df3['Hrs_semana']=df3['Hrs_semana'].fillna(df3['Hrs_semana'].mean())
df3['Balance_vida_trabajo']=df3['Balance_vida_trabajo'].fillna(df3['Balance_vida_trabajo'].mean())
df3['Salud_Mental']=df3['Salud_Mental'].fillna(df3['Salud_Mental'].mode()[0])
df3['Industria']=df3['Industria'].fillna(df3['Industria'].mode()[0])
df3['Recurso_SaludM']=df3['Recurso_SaludM'].fillna(df3['Recurso_SaludM'].mode()[0])
df3['Ubicacion']=df3['Ubicacion'].fillna(df3['Ubicacion'].mode()[0])
df3['Region']=df3['Region'].fillna(df3['Region'].mode()[0])
df3['Cambio_en_productividad']=df3['Cambio_en_productividad'].fillna(df3['Cambio_en_productividad'].mode()[0])
df3['Calificación_aislamiento']=df3['Calificación_aislamiento'].fillna(df3['Calificación_aislamiento'].mode()[0])
df3['Satisfacción_trabajo']=df3['Satisfacción_trabajo'].fillna(df3['Satisfacción_trabajo'].mode()[0])
df3['Apoyo_de_empresa']=df3['Apoyo_de_empresa'].fillna(df3['Apoyo_de_empresa'].mode()[0])
df3['Calidad_del_sueño']=df3['Calidad_del_sueño'].fillna(df3['Calidad_del_sueño'].mode()[0])

df3
```

✓ 0.0s

	Edad	Genero	Rol	Industria	Años_Expe	Ubicacion	Hrs_semana	Balance_vida_trabajo	Nivel_estres	Salud_Mental	Recurso_SaludM	Cambio_en_productividad	Calificación_aislamientoS	Satisfacción_trabajo	Apoyo_de_em
1	40.0	Female	Data Scientist	IT	3.0	Remote	52.0		1.0	Medium	Anxiety	No	Increase	3.0	Satisfied
2	59.0	Non-binary	Software Engineer	Education	22.0	Hybrid	46.0		5.0	Medium	Anxiety	No	No Change	4.0	Unsatisfied
3	27.0	Male	Software Engineer	Finance	20.0	Onsite	32.0		4.0	High	Depression	Yes	Increase	3.0	Unsatisfied
4	49.0	Male	Sales	Consulting	32.0	Onsite	35.0		2.0	High	Anxiety	Yes	Decrease	3.0	Unsatisfied
5	59.0	Non-binary	Sales	IT	31.0	Hybrid	39.0		4.0	High	Anxiety	No	Increase	5.0	Unsatisfied
...
6149	56.0	Prefer not to say	Software Engineer	Education	5.0	Hybrid	35.0		5.0	Medium	Anxiety	Yes	Increase	2.0	Neutral
6151	41.0	Prefer not to say	Project Manager	IT	9.0	Onsite	50.0		2.0	High	Anxiety	Yes	No Change	5.0	Neutral
6152	28.0	Female	Marketing	Retail	5.0	Onsite	52.0		4.0	Low	Burnout	Yes	Increase	1.0	Unsatisfied
6153	58.0	Male	Data Scientist	Education	19.0	Remote	34.0		2.0	Medium	Anxiety	No	Decrease	5.0	Satisfied
6155	59.0	Non-binary	Marketing	Manufacturing	20.0	Onsite	21.0		3.0	High	Burnout	No	Increase	1.0	Neutral

5040 rows x 17 columns

Resultados.

En conclusión, la base de datos ha sido trabajada en un proceso de limpieza, pudimos ver los conflictos que había en ella. Los datos ahora nos permitirán hacer el análisis correcto, sin embargo el análisis pudiera no llegar a ser 100% confiable ya que en el transcurso de limpieza siempre llegan a perderse algunos datos.

Tipo de dato final

```
df3.info()
[121] ✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
Index: 5040 entries, 1 to 6155
Data columns (total 17 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                -
0   Edad                                 5040 non-null   int64
1   Genero                             5040 non-null   object
2   Rol                                 5040 non-null   object
3   Industria                          5040 non-null   object
4   Años_Expe                          5040 non-null   int64
5   Ubicacion                          5040 non-null   object
6   Hrs_semana                         5040 non-null   int64
7   Balance_vida_trabajo               5040 non-null   int64
8   Nivel_estres                      5040 non-null   object
9   Salud_Mental                      5040 non-null   object
10  Recurso_SaludM                    5040 non-null   object
11  Cambio_en_productividad            5040 non-null   object
12  Calificación_aislamientoS          5040 non-null   float64
13  Satisfacción_trabajo               5040 non-null   object
14  Apoyo_de_empresa                  5040 non-null   int64
15  Calidad_del_sueño                  5040 non-null   object
16  Region                            5040 non-null   object
dtypes: float64(1), int64(5), object(11)
memory usage: 708.8+ KB
```

Tabla de datos faltantes, a comparación de la tabla que se presentó al inicio con porcentajes un tanto altos en cada columna, en la actual podemos observar que no existen datos faltantes.

```
#%de datos faltantes
df3.isnull().mean()
[97] ✓ 0.0s

Edad      0.0
Genero    0.0
Rol       0.0
Industria 0.0
Años_Expe 0.0
Ubicacion 0.0
Hrs_semana 0.0
Balance_vida_trabajo 0.0
Nivel_estres 0.0
Salud_Mental 0.0
Recurso_SaludM 0.0
Cambio_en_productividad 0.0
Calificación_aislamientoS 0.0
Satisfacción_trabajo 0.0
Apoyo_de_empresa 0.0
Calidad_del_sueño 0.0
Region    0.0
dtype: float64
```


Datos inválidos

De igual manera los datos 'bbb' ya se encuentran eliminados de la base de datos

```
#cantidad de 'bbb' encontrados despues de a eliminación
for i in lista:
    print(f"En la columna {i} la cadena de 'bbb' son: {df[df[i] == 'bbb'].shape[0]}")

22] ✓ 0.0s

En la columna ID_E la cadena de 'bbb' son: 0
En la columna Edad la cadena de 'bbb' son: 0
En la columna Genero la cadena de 'bbb' son: 0
En la columna Rol la cadena de 'bbb' son: 0
En la columna Industria la cadena de 'bbb' son: 0
En la columna Años_Expe la cadena de 'bbb' son: 0
En la columna Ubicacion la cadena de 'bbb' son: 0
En la columna Hrs_semana la cadena de 'bbb' son: 0
En la columna N_reun_V la cadena de 'bbb' son: 0
En la columna Balance_vida_trabajo la cadena de 'bbb' son: 0
En la columna Nivel_estres la cadena de 'bbb' son: 0
En la columna Salud_Mental la cadena de 'bbb' son: 0
En la columna Recurso_SaludM la cadena de 'bbb' son: 0
En la columna Cambio_en_productividad la cadena de 'bbb' son: 0
En la columna Calificación_aislamientos la cadena de 'bbb' son: 0
En la columna Satisfacción_trabajo la cadena de 'bbb' son: 0
En la columna Apoyo_de_empresa la cadena de 'bbb' son: 0
En la columna Actividad_fisica la cadena de 'bbb' son: 0
En la columna Calidad_del_sueño la cadena de 'bbb' son: 0
En la columna Region la cadena de 'bbb' son: 0
```