configuración del entorno

completar líneas

In [247…

```
%config Completer.use_jedi = True
```

Obtener dataset

Instalar kagglehub

In [1]:

```
conda install kagglehub
```

```
Channels:
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): ...working... done
Solving environment: ...working... done

## Package Plan ##

  environment location: C:\Users\darly\anaconda3\envs\IAexplores

  added / updated specs:
    - kagglehub


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    tqdm-4.67.1                |  py312hfc267ef_0         187 KB
    ------------------------------------------------------------
                                           Total:         187 KB

The following NEW packages will be INSTALLED:

  kagglehub          pkgs/main/win-64::kagglehub-0.2.7-py312haa95532_0
  tqdm               pkgs/main/win-64::tqdm-4.67.1-py312hfc267ef_0



Downloading and Extracting Packages: ...working...
tqdm-4.67.1          | 187 KB    |            |   0%
tqdm-4.67.1          | 187 KB    | 8          |   9%
tqdm-4.67.1          | 187 KB    | ######8    |  68%
tqdm-4.67.1          | 187 KB    | ########## | 100%
tqdm-4.67.1          | 187 KB    | ########## | 100%

 done
Preparing transaction: done
Executing transaction: done

Note: you may need to restart the kernel to use updated packages.
```

importa kaggle, pandas y numpy , y descargar data

In [56]:

```python
import kagglehub  #descargar dataset
import pandas as pd #procesos de tabla
import numpy as np  #procesos de vectores y matemáticas

#visualizacion
import plotly.express as px
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
```

In [2]:

```python
# Download latest version
path = kagglehub.dataset_download("ruchi798/data-science-job-salaries")
```

```
print("Path to dataset files:", path)
```

```
C:\Users\darly\anaconda3\envs\IAexplores\Lib\site-packages\tqdm\auto.py:21: TqdmWarn
ing: IProgress not found. Please update jupyter and ipywidgets. See https://ipywidge
ts.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
Warning: Looks like you're using an outdated `kagglehub` version, please consider up
dating (latest version: 0.3.10)
Downloading from https://www.kaggle.com/api/v1/datasets/download/ruchi798/data-scien
ce-job-salaries?dataset_version_number=1...
100%|██████████| 7.37k/7.37k [00:00<?, ?B/s]
Extracting model files...
Path to dataset files: C:\Users\darly\.cache\kagglehub\datasets\ruchi798\data-scienc
e-job-salaries\versions\1
```

crear un data frame, una tabla como ejemplo

In [6]:

```
data= pd.DataFrame({
    "nombres": ["ana", "juana", "sara"],
    "edad": [12,23,34]
})
data
```

Out[6]:

	nombres	edad
0	ana	12
1	juana	23
2	sara	34

In [7]:

```
data2= pd.DataFrame({
    "nombres": ["ana", "juana", "sara"],
    "salario": [120,230,340]
})
data2
```

Out[7]:

	nombres	salario
0	ana	120
1	juana	230
2	sara	340

unir data_frame

In [8]:

```
new_df= data.merge(data2)
```

In [9]:

```
new_df
```

Out[9]:

	nombres	edad	salario
0	ana	12	120
1	juana	23	230
2	sara	34	340

leer un archivo csv, ya descargado, e imprimir la cabeza (primero 5 elementos)

In [130…

```
df = pd.read_csv("C:/Users/darly/.cache/kagglehub/datasets/ruchi798/data-science-jo
```

In [131…

```
df = pd.read_csv(r"C:\Users\darly\.cache\kagglehub\datasets\ruchi798\data-science-j
```

exploración, filtro y limpieza de la data

mostrar las primeras 5 filas

In [132…

```
df.head()
```

Out[132…

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_curr
0	0	2020	MI	FT	Data Scientist	70000	
1	1	2020	SE	FT	Machine Learning Scientist	260000	
2	2	2020	SE	FT	Big Data Engineer	85000	
3	3	2020	MI	FT	Product Data Analyst	20000	
4	4	2020	SE	FT	Machine Learning Engineer	150000	

mostrar las últimas 5 lineas

In [134…

```
df.tail()
```

Out[134…

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_cu
602	602	2022	SE	FT	Data Engineer	154000	
603	603	2022	SE	FT	Data Engineer	126000	
604	604	2022	SE	FT	Data Analyst	129000	
605	605	2022	SE	FT	Data Analyst	150000	
606	606	2022	MI	FT	AI Scientist	200000	

para describir la data, muestra un resumen del dataset solo en las variables numericas

In [135…

```
df.describe()
```

Out[135…

	Unnamed: 0	work_year	salary	salary_in_usd	remote_ratio
count	607.000000	607.000000	6.070000e+02	607.000000	607.00000
mean	303.000000	2021.405272	3.240001e+05	112297.869852	70.92257
std	175.370085	0.692133	1.544357e+06	70957.259411	40.70913
min	0.000000	2020.000000	4.000000e+03	2859.000000	0.00000
25%	151.500000	2021.000000	7.000000e+04	62726.000000	50.00000
50%	303.000000	2022.000000	1.150000e+05	101570.000000	100.00000
75%	454.500000	2022.000000	1.650000e+05	150000.000000	100.00000
max	606.000000	2022.000000	3.040000e+07	600000.000000	100.00000

muestra una lista con todas las columnas que tiene el data frame

In [136…

```
df.columns
```

Out[136…

```
Index(['Unnamed: 0', 'work_year', 'experience_level', 'employment_type',
       'job_title', 'salary', 'salary_currency', 'salary_in_usd',
       'employee_residence', 'remote_ratio', 'company_location',
       'company_size'],
      dtype='object')
```

esto sirve para hacer consultas especificas del dataframe

In [137…

```
df[df.salary_in_usd > 250000]
```

Out[137…

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_cu
1	1	2020	SE	FT	Machine Learning Scientist	260000	
25	25	2020	EX	FT	Director of Data Science	325000	
33	33	2020	MI	FT	Research Scientist	450000	
63	63	2020	SE	FT	Data Scientist	412000	
78	78	2021	MI	CT	ML Engineer	270000	
93	93	2021	SE	FT	Lead Data Engineer	276000	
97	97	2021	MI	FT	Financial Data Analyst	450000	
157	157	2021	MI	FT	Applied Machine Learning Scientist	423000	
225	225	2021	EX	CT	Principal Data Scientist	416000	
231	231	2021	SE	FT	ML Engineer	256000	
252	252	2021	EX	FT	Principal Data Engineer	600000	
416	416	2022	SE	FT	Data Scientist	260000	
482	482	2022	EX	FT	Data Engineer	324000	
519	519	2022	SE	FT	Applied Data Scientist	380000	
523	523	2022	SE	FT	Data Analytics Lead	405000	

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_cu
534	534	2022	SE	FT	Data Architect	266400	

In [138…

```
df[df.salary_in_usd > 250000].describe()
```

Out[138…

	Unnamed: 0	work_year	salary	salary_in_usd	remote_ratio
count	16.00000	16.000000	16.000000	16.000000	16.000000
mean	233.06250	2021.062500	360837.500000	360837.500000	78.125000
std	197.70364	0.771902	97733.221066	97733.221066	40.697051
min	1.00000	2020.000000	256000.000000	256000.000000	0.000000
25%	74.25000	2020.750000	269100.000000	269100.000000	87.500000
50%	191.00000	2021.000000	352500.000000	352500.000000	100.000000
75%	432.50000	2022.000000	417750.000000	417750.000000	100.000000
max	534.00000	2022.000000	600000.000000	600000.000000	100.000000

realizar consulta para datos cualitativos

In [139…

```
df.job_title
```

Out[139…

```
0                   Data Scientist
1        Machine Learning Scientist
2                Big Data Engineer
3             Product Data Analyst
4        Machine Learning Engineer
                   ...
602                  Data Engineer
603                  Data Engineer
604                   Data Analyst
605                   Data Analyst
606                  AI Scientist
Name: job_title, Length: 607, dtype: object
```

In [140…

```
df.query("job_title == 'Data Scientist'") #RECUERDE QUE LA CONSULTA QUERY DEBE SER
```

Out[140…

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_
0	0	2020	MI	FT	Data Scientist	70000	
7	7	2020	MI	FT	Data Scientist	11000000	
10	10	2020	EN	FT	Data Scientist	45000	
11	11	2020	MI	FT	Data Scientist	3000000	
12	12	2020	EN	FT	Data Scientist	35000	
...	...	...	...	...	...	...	
592	592	2022	SE	FT	Data Scientist	230000	
593	593	2022	SE	FT	Data Scientist	150000	
596	596	2022	SE	FT	Data Scientist	210000	
598	598	2022	MI	FT	Data Scientist	160000	
599	599	2022	MI	FT	Data Scientist	130000	

143 rows × 12 columns

las filas determinadas

In [141…

```
df.iloc[20:40]
```

Out[141…

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_
20	20	2020	MI	FT	Machine Learning Engineer	299000	
21	21	2020	MI	FT	Product Data Analyst	450000	
22	22	2020	SE	FT	Data Engineer	42000	
23	23	2020	MI	FT	BI Data Analyst	98000	
24	24	2020	MI	FT	Lead Data Scientist	115000	
25	25	2020	EX	FT	Director of Data Science	325000	
26	26	2020	EN	FT	Research Scientist	42000	
27	27	2020	SE	FT	Data Engineer	720000	
28	28	2020	EN	CT	Business Data Analyst	100000	
29	29	2020	SE	FT	Machine Learning Manager	157000	
30	30	2020	MI	FT	Data Engineering Manager	51999	
31	31	2020	EN	FT	Big Data Engineer	70000	
32	32	2020	SE	FT	Data Scientist	60000	
33	33	2020	MI	FT	Research Scientist	450000	
34	34	2020	MI	FT	Data Analyst	41000	
35	35	2020	MI	FT	Data Engineer	65000	

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_
36	36	2020	MI	FT	Data Science Consultant	103000	
37	37	2020	EN	FT	Machine Learning Engineer	250000	
38	38	2020	EN	FT	Data Analyst	10000	
39	39	2020	EN	FT	Machine Learning Engineer	138000	

columnas específicas de una dataframe

In [142…

```
df[["job_title", "salary"]]
```

Out[142…

	job_title	salary
0	Data Scientist	70000
1	Machine Learning Scientist	260000
2	Big Data Engineer	85000
3	Product Data Analyst	20000
4	Machine Learning Engineer	150000
...	...	...
602	Data Engineer	154000
603	Data Engineer	126000
604	Data Analyst	129000
605	Data Analyst	150000
606	AI Scientist	200000

607 rows × 2 columns

otra forma es con la estructura iloc, pero no dando nombres sino posiciones 8recordar que la primera posicion es filas las demas columnas)

In [143…

```
df.iloc[:, [2,4,5]]
```

Out[143…

	experience_level	job_title	salary
0	MI	Data Scientist	70000
1	SE	Machine Learning Scientist	260000
2	SE	Big Data Engineer	85000
3	MI	Product Data Analyst	20000
4	SE	Machine Learning Engineer	150000
...	...	...	...
602	SE	Data Engineer	154000
603	SE	Data Engineer	126000
604	SE	Data Analyst	129000
605	SE	Data Analyst	150000
606	MI	AI Scientist	200000

607 rows × 3 columns

columnas determinadas y filas determinadas (estas ultimas son las primeras)

In [144…

```
df.iloc[10:40, [2,4,5]]
```

Out[144…

	experience_level	job_title	salary
10	EN	Data Scientist	45000
11	MI	Data Scientist	3000000
12	EN	Data Scientist	35000
13	MI	Lead Data Analyst	87000
14	MI	Data Analyst	85000
15	MI	Data Analyst	8000
16	EN	Data Engineer	4450000
17	SE	Big Data Engineer	100000
18	EN	Data Science Consultant	423000
19	MI	Lead Data Engineer	56000
20	MI	Machine Learning Engineer	299000
21	MI	Product Data Analyst	450000
22	SE	Data Engineer	42000
23	MI	BI Data Analyst	98000
24	MI	Lead Data Scientist	115000
25	EX	Director of Data Science	325000
26	EN	Research Scientist	42000
27	SE	Data Engineer	720000
28	EN	Business Data Analyst	100000
29	SE	Machine Learning Manager	157000
30	MI	Data Engineering Manager	51999
31	EN	Big Data Engineer	70000
32	SE	Data Scientist	60000
33	MI	Research Scientist	450000
34	MI	Data Analyst	41000
35	MI	Data Engineer	65000
36	MI	Data Science Consultant	103000
37	EN	Machine Learning Engineer	250000
38	EN	Data Analyst	10000
39	EN	Machine Learning Engineer	138000

las columnas con nombres y no por posicion, desde una a otra

In [145…
```
df.loc[:,"experience_level": "job_title"]
```

Out[145…

	experience_level	employment_type	job_title
0	MI	FT	Data Scientist
1	SE	FT	Machine Learning Scientist
2	SE	FT	Big Data Engineer
3	MI	FT	Product Data Analyst
4	SE	FT	Machine Learning Engineer
...	...	...	...
602	SE	FT	Data Engineer
603	SE	FT	Data Engineer
604	SE	FT	Data Analyst
605	SE	FT	Data Analyst
606	MI	FT	AI Scientist

607 rows × 3 columns

otra forma de consultar, parecido al query

In [146…
```
df.loc[df["experience_level"]== "MI"]
```

Out[146…

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary
0	0	2020	MI	FT	Data Scientist	70000	
3	3	2020	MI	FT	Product Data Analyst	20000	
7	7	2020	MI	FT	Data Scientist	11000000	
8	8	2020	MI	FT	Business Data Analyst	135000	
11	11	2020	MI	FT	Data Scientist	3000000	
...	...	...	...	...	...	...	
567	567	2022	MI	FT	Data Analyst	50000	
586	586	2022	MI	FT	Data Analyst	35000	
598	598	2022	MI	FT	Data Scientist	160000	
599	599	2022	MI	FT	Data Scientist	130000	
606	606	2022	MI	FT	AI Scientist	200000	

213 rows × 12 columns

In [147…

```
df.loc[df["experience_level"]== "MI", ["job_title",     "salary"]]
```

Out[147…

	job_title	salary
0	Data Scientist	70000
3	Product Data Analyst	20000
7	Data Scientist	11000000
8	Business Data Analyst	135000
11	Data Scientist	3000000
...	...	...
567	Data Analyst	50000
586	Data Analyst	35000
598	Data Scientist	160000
599	Data Scientist	130000
606	AI Scientist	200000

213 rows × 2 columns

In [148…

```
df.loc[df["experience_level"]== "MI", ["job_title",     "salary"]].sort_values("sal
```

Out[148…

	job_title	salary
185	Data Engineer	4000
15	Data Analyst	8000
184	Machine Learning Scientist	12000
192	Big Data Engineer	18000
208	Data Engineer	20000
...	...	...
136	ML Engineer	7000000
137	ML Engineer	8500000
7	Data Scientist	11000000
102	BI Data Analyst	11000000
177	Data Scientist	30400000

213 rows × 2 columns

cambiar el nombre de una columna

In [149…

```
df.rename(columns= {"salary": "salario"})
```

Out[149…

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salario	salary_cu
0	0	2020	MI	FT	Data Scientist	70000	
1	1	2020	SE	FT	Machine Learning Scientist	260000	
2	2	2020	SE	FT	Big Data Engineer	85000	
3	3	2020	MI	FT	Product Data Analyst	20000	
4	4	2020	SE	FT	Machine Learning Engineer	150000	
...	...	...	...	...	...	...	
602	602	2022	SE	FT	Data Engineer	154000	
603	603	2022	SE	FT	Data Engineer	126000	
604	604	2022	SE	FT	Data Analyst	129000	
605	605	2022	SE	FT	Data Analyst	150000	
606	606	2022	MI	FT	AI Scientist	200000	

607 rows × 12 columns

borrar columnas

In [150…

```
df.drop(columns={"salary"})
```

Out[150…

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary_currency
0	0	2020	MI	FT	Data Scientist	EUR
1	1	2020	SE	FT	Machine Learning Scientist	USD
2	2	2020	SE	FT	Big Data Engineer	GBP
3	3	2020	MI	FT	Product Data Analyst	USD
4	4	2020	SE	FT	Machine Learning Engineer	USD
...	...	...	...	...	...	...
602	602	2022	SE	FT	Data Engineer	USD
603	603	2022	SE	FT	Data Engineer	USD
604	604	2022	SE	FT	Data Analyst	USD
605	605	2022	SE	FT	Data Analyst	USD
606	606	2022	MI	FT	AI Scientist	USD

607 rows × 11 columns

agregar una nueva columna o modificarla

In [151…

```
df["salario en pesos"] = df.salary * 4500
df
```

Out[151…

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_cu
0	0	2020	MI	FT	Data Scientist	70000	
1	1	2020	SE	FT	Machine Learning Scientist	260000	
2	2	2020	SE	FT	Big Data Engineer	85000	
3	3	2020	MI	FT	Product Data Analyst	20000	
4	4	2020	SE	FT	Machine Learning Engineer	150000	
...	...	...	...	...	...	...	
602	602	2022	SE	FT	Data Engineer	154000	
603	603	2022	SE	FT	Data Engineer	126000	
604	604	2022	SE	FT	Data Analyst	129000	
605	605	2022	SE	FT	Data Analyst	150000	
606	606	2022	MI	FT	AI Scientist	200000	

607 rows × 13 columns

obtener muestras aleatorias (usos testing)

In [152…

```
df.sample(frac=0.5) #fragmento deel 50 por ciento de los datos
```

Out[152…

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_
488	488	2022	MI	FL	Data Scientist	100000	
427	427	2022	MI	FT	Data Engineer	45000	
92	92	2021	MI	FT	Lead Data Analyst	1450000	
500	500	2022	SE	FT	Machine Learning Engineer	57000	
179	179	2021	MI	FT	Data Scientist	420000	
...	...	...	...	...	...	...	
606	606	2022	MI	FT	AI Scientist	200000	
192	192	2021	MI	FT	Big Data Engineer	18000	
491	491	2022	MI	FT	Principal Data Analyst	75000	
123	123	2021	EN	FT	Applied Data Scientist	80000	
82	82	2021	MI	FT	Applied Data Scientist	68000	

304 rows × 13 columns

In [153…

```
df.sample(n=100) #numero determinado de muestras
```

Out[153…

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_cu
100	100	2021	MI	FT	Data Analyst	75000	
88	88	2021	SE	FT	Lead Data Analyst	170000	
382	382	2022	SE	FT	Data Analyst	128875	
149	149	2021	SE	FT	Cloud Data Engineer	160000	
507	507	2022	MI	FT	Research Scientist	59000	
...	...	...	...	...	...	...	
316	316	2022	EN	FT	Data Engineer	35000	
115	115	2021	EN	FT	Machine Learning Scientist	225000	
179	179	2021	MI	FT	Data Scientist	420000	
4	4	2020	SE	FT	Machine Learning Engineer	150000	
499	499	2022	EN	FT	Data Scientist	66500	

100 rows × 13 columns

agrupar datos determinados y bajo una medida

In [154…

```
df.groupby("job_title").mean(numeric_only=True)
```

Out[154…

	Unnamed: 0	work_year	salary	salary_in_usd	remote_ratio	sala
job_title						
3D Computer Vision Researcher	77.000000	2021.000000	4.000000e+05	5409.000000	50.000000	1.80000
AI Scientist	254.142857	2021.142857	2.905714e+05	66135.571429	78.571429	1.30757
Analytics Engineer	458.250000	2022.000000	1.750000e+05	175000.000000	50.000000	7.87500
Applied Data Scientist	351.600000	2021.600000	1.724000e+05	175655.000000	70.000000	7.75800
Applied Machine Learning Scientist	321.000000	2021.500000	1.413500e+05	142068.750000	87.500000	6.36075
BI Data Analyst	106.333333	2020.833333	1.902045e+06	74755.166667	66.666667	8.55920
Big Data Architect	255.000000	2021.000000	1.250000e+05	99703.000000	50.000000	5.62500
Big Data Engineer	123.125000	2020.625000	4.550000e+05	51974.000000	50.000000	2.04750
Business Data Analyst	256.800000	2021.000000	3.550000e+05	76691.200000	90.000000	1.59750
Cloud Data Engineer	122.000000	2021.000000	1.400000e+05	124647.000000	75.000000	6.30000
Computer Vision Engineer	274.833333	2021.166667	8.350000e+04	44419.333333	58.333333	3.75750
Computer Vision Software Engineer	235.666667	2021.333333	1.003333e+05	105248.666667	100.000000	4.51500
Data Analyst	362.010309	2021.680412	9.660496e+04	92893.061856	75.257732	4.34722
Data Analytics Engineer	216.750000	2021.250000	6.175000e+04	64799.250000	75.000000	2.77875
Data Analytics Lead	523.000000	2022.000000	4.050000e+05	405000.000000	100.000000	1.82250
Data Analytics Manager	366.285714	2021.571429	1.271343e+05	127134.285714	85.714286	5.72104

	Unnamed: 0	work_year	salary	salary_in_usd	remote_ratio	sala
job_title						
Data Architect	390.636364	2021.727273	1.778739e+05	177873.909091	100.000000	8.00432
Data Engineer	343.537879	2021.590909	1.792106e+05	112725.000000	75.000000	8.06447
Data Engineering Manager	107.200000	2020.600000	1.197998e+05	123227.200000	70.000000	5.39099
Data Science Consultant	138.000000	2020.714286	1.227143e+05	69420.714286	71.428571	5.52214
Data Science Engineer	229.666667	2021.333333	8.450000e+04	75803.333333	83.333333	3.80250
Data Science Manager	274.000000	2021.333333	1.062599e+06	158328.500000	83.333333	4.78169
Data Scientist	314.832168	2021.391608	5.083472e+05	108187.832168	63.986014	2.28756
Data Specialist	165.000000	2021.000000	1.650000e+05	165000.000000	100.000000	7.42500
Director of Data Engineering	171.500000	2021.000000	1.412500e+05	156738.000000	100.000000	6.35625
Director of Data Science	185.857143	2021.000000	1.932857e+05	195074.000000	42.857143	8.69785
ETL Developer	373.500000	2022.000000	5.000000e+04	54957.000000	0.000000	2.25000
Finance Data Analyst	183.000000	2021.000000	4.500000e+04	61896.000000	50.000000	2.02500
Financial Data Analyst	279.000000	2021.500000	2.750000e+05	275000.000000	75.000000	1.23750
Head of Data	302.200000	2021.400000	1.564000e+05	160162.600000	90.000000	7.03800
Head of Data Science	270.250000	2021.500000	1.467188e+05	146718.750000	50.000000	6.60234
Head of Machine Learning	384.000000	2022.000000	6.000000e+06	79039.000000	50.000000	2.70000
Lead Data Analyst	64.333333	2020.666667	5.690000e+05	92203.000000	100.000000	2.56050
Lead Data Engineer	145.500000	2020.833333	1.403333e+05	139724.500000	66.666667	6.31500

	Unnamed: 0	work_year	salary	salary_in_usd	remote_ratio	sala
job_title						
Lead Data Scientist	53.000000	2020.333333	1.101667e+06	115190.000000	50.000000	4.95750
Lead Machine Learning Engineer	457.000000	2022.000000	8.000000e+04	87932.000000	0.000000	3.60000
ML Engineer	179.333333	2021.000000	2.676667e+06	117504.000000	83.333333	1.20450
Machine Learning Developer	358.000000	2021.666667	1.000000e+05	85860.666667	83.333333	4.50000
Machine Learning Engineer	288.585366	2021.317073	2.727179e+05	104880.146341	67.073171	1.22723
Machine Learning Infrastructure Engineer	234.333333	2021.000000	9.733333e+04	101145.000000	50.000000	4.38000
Machine Learning Manager	29.000000	2020.000000	1.570000e+05	117104.000000	50.000000	7.06500
Machine Learning Scientist	248.000000	2021.250000	1.584125e+05	158412.500000	68.750000	7.12856
Marketing Data Analyst	90.000000	2021.000000	7.500000e+04	88654.000000	100.000000	3.37500
NLP Engineer	455.000000	2022.000000	2.400000e+05	37236.000000	50.000000	1.08000
Principal Data Analyst	370.000000	2021.500000	1.225000e+05	122500.000000	100.000000	5.51250
Principal Data Engineer	196.000000	2021.000000	3.283333e+05	328333.333333	100.000000	1.47750
Principal Data Scientist	205.285714	2021.000000	2.067143e+05	215242.428571	85.714286	9.30214
Product Data Analyst	12.000000	2020.000000	2.350000e+05	13036.000000	50.000000	1.05750
Research Scientist	246.562500	2021.125000	1.104937e+05	109019.500000	53.125000	4.97221
Staff Data Scientist	283.000000	2021.000000	1.050000e+05	105000.000000	100.000000	4.72500

In [155…

```
df.groupby("job_title").mean(numeric_only=True).count() #cuenta
```

Out[155…

```
Unnamed: 0          50
work_year           50
salary              50
salary_in_usd       50
remote_ratio        50
salario en pesos    50
dtype: int64
```

In [156…

```
df.groupby("job_title").agg({
    "salary": ["max", "mean"]
})  #agrupar por una columan y determinadas medidas
```

Out[156…

	salary	
	max	mean
job_title		
3D Computer Vision Researcher	400000	4.000000e+05
AI Scientist	1335000	2.905714e+05
Analytics Engineer	205300	1.750000e+05
Applied Data Scientist	380000	1.724000e+05
Applied Machine Learning Scientist	423000	1.413500e+05
BI Data Analyst	11000000	1.902045e+06
Big Data Architect	125000	1.250000e+05
Big Data Engineer	1672000	4.550000e+05
Business Data Analyst	1400000	3.550000e+05
Cloud Data Engineer	160000	1.400000e+05
Computer Vision Engineer	180000	8.350000e+04
Computer Vision Software Engineer	150000	1.003333e+05
Data Analyst	450000	9.660496e+04
Data Analytics Engineer	110000	6.175000e+04
Data Analytics Lead	405000	4.050000e+05
Data Analytics Manager	150260	1.271343e+05
Data Architect	266400	1.778739e+05
Data Engineer	4450000	1.792106e+05
Data Engineering Manager	174000	1.197998e+05
Data Science Consultant	423000	1.227143e+05
Data Science Engineer	159500	8.450000e+04
Data Science Manager	7000000	1.062599e+06
Data Scientist	30400000	5.083472e+05
Data Specialist	165000	1.650000e+05
Director of Data Engineering	200000	1.412500e+05
Director of Data Science	325000	1.932857e+05
ETL Developer	50000	5.000000e+04
Finance Data Analyst	45000	4.500000e+04

	salary	
	max	mean
job_title		
Financial Data Analyst	450000	2.750000e+05
Head of Data	235000	1.564000e+05
Head of Data Science	224000	1.467188e+05
Head of Machine Learning	6000000	6.000000e+06
Lead Data Analyst	1450000	5.690000e+05
Lead Data Engineer	276000	1.403333e+05
Lead Data Scientist	3000000	1.101667e+06
Lead Machine Learning Engineer	80000	8.000000e+04
ML Engineer	8500000	2.676667e+06
Machine Learning Developer	100000	1.000000e+05
Machine Learning Engineer	4900000	2.727179e+05
Machine Learning Infrastructure Engineer	195000	9.733333e+04
Machine Learning Manager	157000	1.570000e+05
Machine Learning Scientist	260000	1.584125e+05
Marketing Data Analyst	75000	7.500000e+04
NLP Engineer	240000	2.400000e+05
Principal Data Analyst	170000	1.225000e+05
Principal Data Engineer	600000	3.283333e+05
Principal Data Scientist	416000	2.067143e+05
Product Data Analyst	450000	2.350000e+05
Research Scientist	450000	1.104937e+05
Staff Data Scientist	105000	1.050000e+05

contar elementos de una columnas

In [157…

```
df.shape #tamaño de data
```

Out[157…

```
(607, 13)
```

elementos unicos de cada columna

In [158…

```
df.nunique()
```

Out[158…

```
Unnamed: 0            607
work_year               3
experience_level        4
employment_type         4
job_title              50
salary                272
salary_currency        17
salary_in_usd         369
employee_residence     57
remote_ratio            3
company_location       50
company_size            3
salario en pesos      272
dtype: int64
```

hacer limpieza de datos

In [159…

```
df.count() #contar datos
```

Out[159…

```
Unnamed: 0            607
work_year             607
experience_level      607
employment_type       607
job_title             607
salary                607
salary_currency       607
salary_in_usd         607
employee_residence    607
remote_ratio          607
company_location      607
company_size          607
salario en pesos      607
dtype: int64
```

In [160…

```
df.isnull().sum() #que datos son nulos
```

Out[160…

```
Unnamed: 0            0
work_year             0
experience_level      0
employment_type       0
job_title             0
salary                0
salary_currency       0
salary_in_usd         0
employee_residence    0
remote_ratio          0
company_location      0
company_size          0
salario en pesos      0
dtype: int64
```

Visualizacion de la data a partir de gráficos

In [161…

```
top10_job_title = df['job_title'].value_counts()[:10] #las primeras 10 empleos mas
```

dibujar un diagrama de barras * px.bar(...): Crea un gráfico de barras. * x=top10_job_title.index: Usa los títulos de trabajo (índices de la serie) como el eje X. * y=top10_job_title.values: Usa la cantidad de veces que aparecen los títulos como eje Y. * color=top10_job_title.index: Asigna diferentes colores a cada categoría (título de trabajo). * color_discrete_sequence=px.colors.sequential.PuBuGn: Usa una paleta de colores predefinida (PuBuGn). * text=top10_job_title.values: Muestra los valores sobre las barras. * title='2.1.2. Top 10 Job Titles': Agrega un título al gráfico. * template='plotly_dark': Usa un tema oscuro para el diseño.

In [162…

```
fig = px.bar(y=top10_job_title.values,
             x=top10_job_title.index,
             color = top10_job_title.index,
             color_discrete_sequence=px.colors.sequential.PuBuGn,
             text=top10_job_title.values,
             title= '2.1.2. Top 10 Job Titles',
             template= 'plotly_dark')
fig.show()
```

El método update_layout() se usa para modificar el diseño del gráfico. Aquí está lo que hace cada argumento: * xaxis_title="Job Titles" : Cambia el título del eje X a "Job Titles" (Títulos de Trabajo). y Este eje representa las categorías (diferentes títulos de trabajo). *yaxis_title="count" : Cambia el título del eje Y a "count" (Cantidad). Este eje muestra la frecuencia de cada título de trabajo en los datos. * font=dict(size=17, family="Franklin Gothic") Ajusta el tamaño y la fuente del texto en el gráfico. size=17: Aumenta el tamaño del texto a 17 puntos. family="Franklin Gothic": Usa la fuente "Franklin Gothic" para los textos.

In [163…

```
fig.update_layout(
    xaxis_title="Job Titles",
    yaxis_title="count",
    font = dict(size=17,family="Franklin Gothic"))
fig.show()
```

vamos a construir un digrama de lineas por cada variable cuantitativa, sirve para ver el comportramiento de una variable en el tiempo

In [165…

```
df_cuant= df.select_dtypes(include=['int64', 'float64'])
df_cuant
```

Out[165…

	Unnamed: 0	work_year	salary	salary_in_usd	remote_ratio	salario en pesos
0	0	2020	70000	79833	0	315000000
1	1	2020	260000	260000	0	1170000000
2	2	2020	85000	109024	50	382500000
3	3	2020	20000	20000	0	90000000
4	4	2020	150000	150000	50	675000000
...	...	...	...	...	...	...
602	602	2022	154000	154000	100	693000000
603	603	2022	126000	126000	100	567000000
604	604	2022	129000	129000	0	580500000
605	605	2022	150000	150000	100	675000000
606	606	2022	200000	200000	100	900000000

607 rows × 6 columns

In [166…

```
df_cuant= df_cuant.iloc[:, 1:]
```

In [167…

```
df_cuant
```

Out[167…

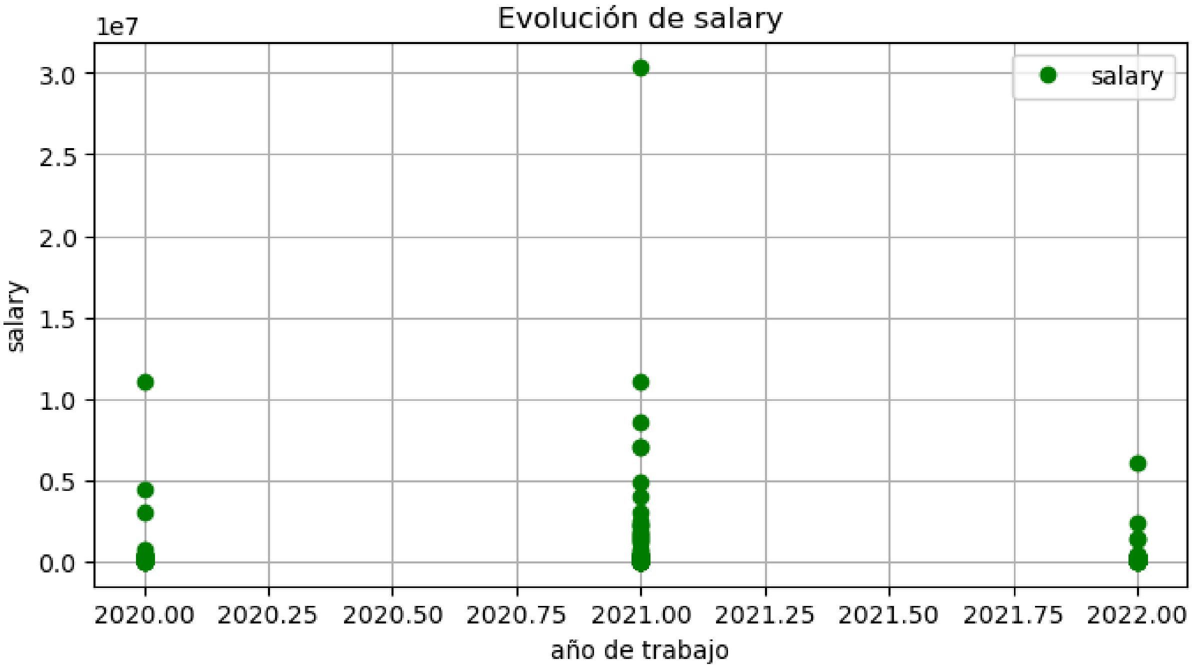
	work_year	salary	salary_in_usd	remote_ratio	salario en pesos
0	2020	70000	79833	0	315000000
1	2020	260000	260000	0	1170000000
2	2020	85000	109024	50	382500000
3	2020	20000	20000	0	90000000
4	2020	150000	150000	50	675000000
...	...	...	...	...	...
602	2022	154000	154000	100	693000000
603	2022	126000	126000	100	567000000
604	2022	129000	129000	0	580500000
605	2022	150000	150000	100	675000000
606	2022	200000	200000	100	900000000

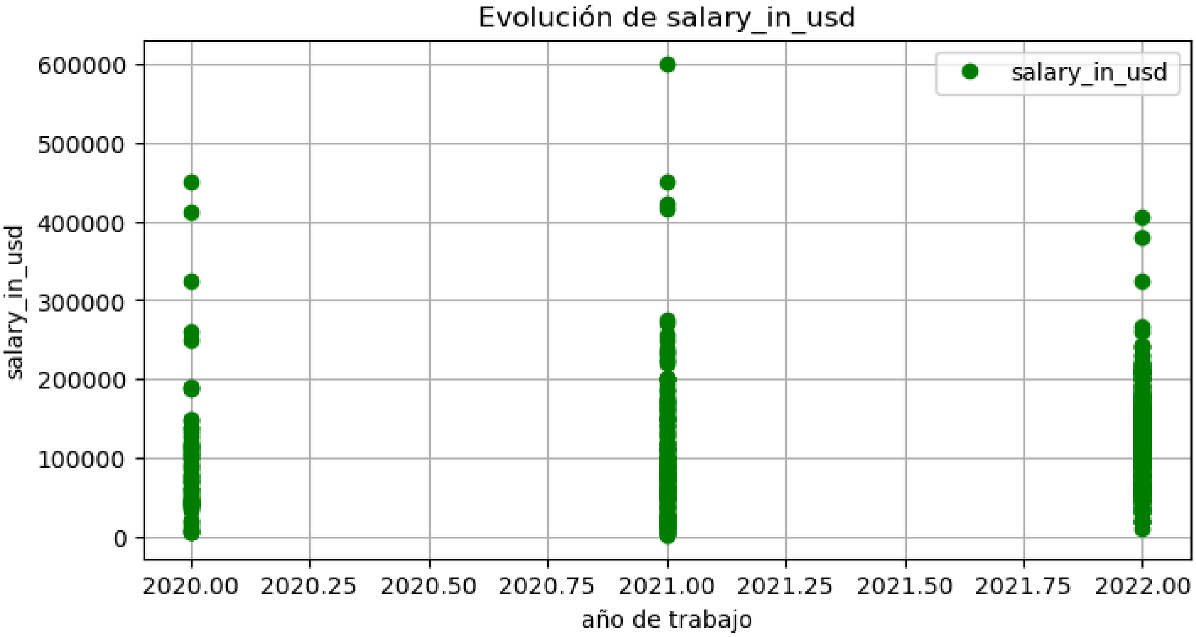
607 rows × 5 columns

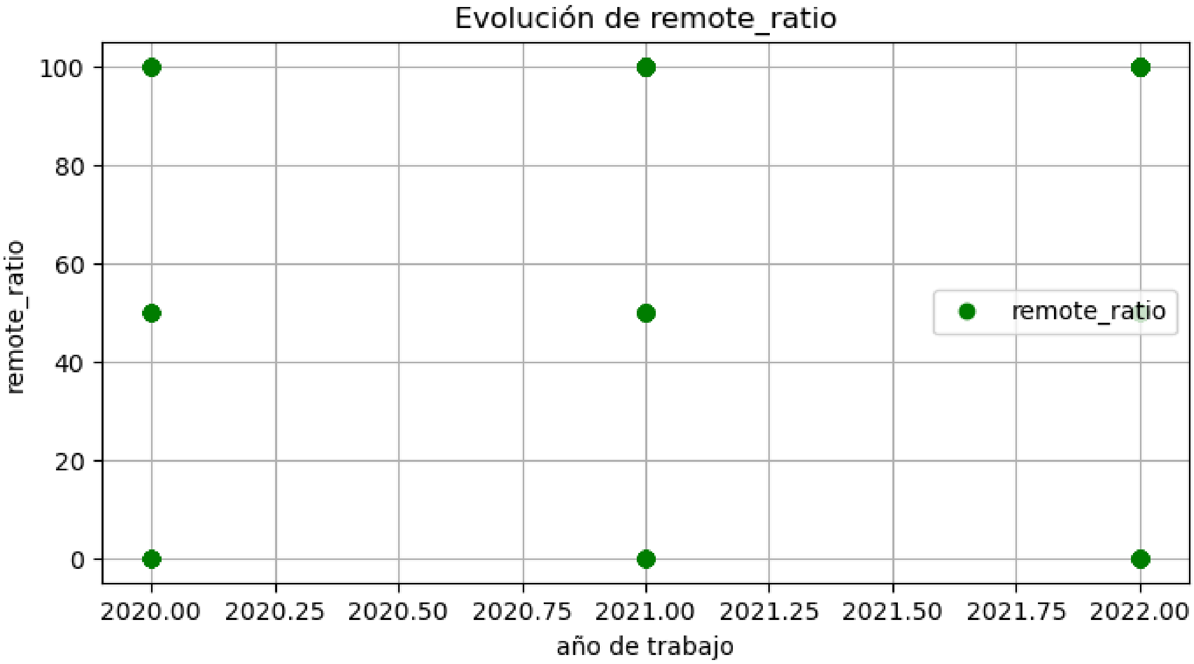
Gráficar uno por uno

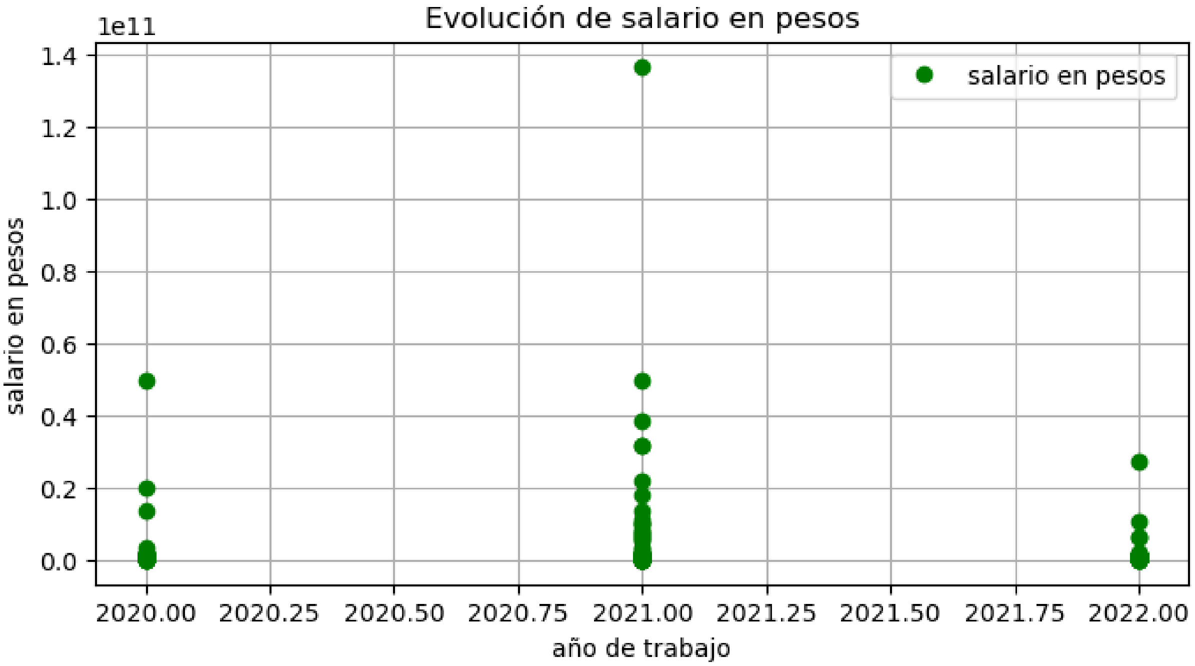
In [168…

```
for i in range(1, df_cuant.shape[1]):
    plt.figure(figsize=(8, 4))  # Crear una nueva figura para cada gráfico

    plt.plot(df_cuant.work_year, df_cuant.iloc[:, i], marker="o", linestyle="",colo

    # Personalización del gráfico
    plt.xlabel("año de trabajo")
    plt.ylabel(df_cuant.columns[i])
    plt.title(f"Evolución de {df_cuant.columns[i]}")
    plt.legend()
    plt.grid(True)

    plt.show()  # Mostrar cada gráf
```









distribución normal

```
In [169…  df_cuant= df_cuant.iloc[:,1:]
          df_cuant
```

Out[169…

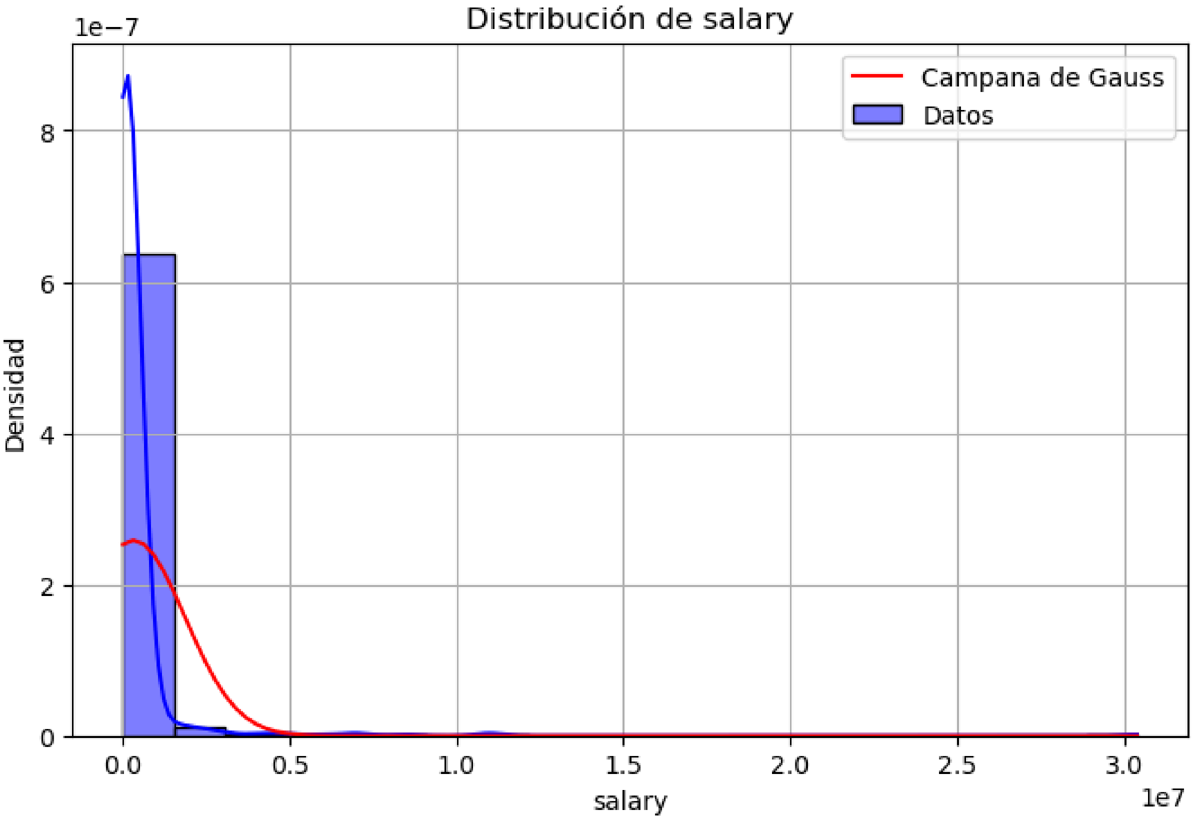
	salary	salary_in_usd	remote_ratio	salario en pesos
0	70000	79833	0	315000000
1	260000	260000	0	1170000000
2	85000	109024	50	382500000
3	20000	20000	0	90000000
4	150000	150000	50	675000000
...	...	...	...	...
602	154000	154000	100	693000000
603	126000	126000	100	567000000
604	129000	129000	0	580500000
605	150000	150000	100	675000000
606	200000	200000	100	900000000

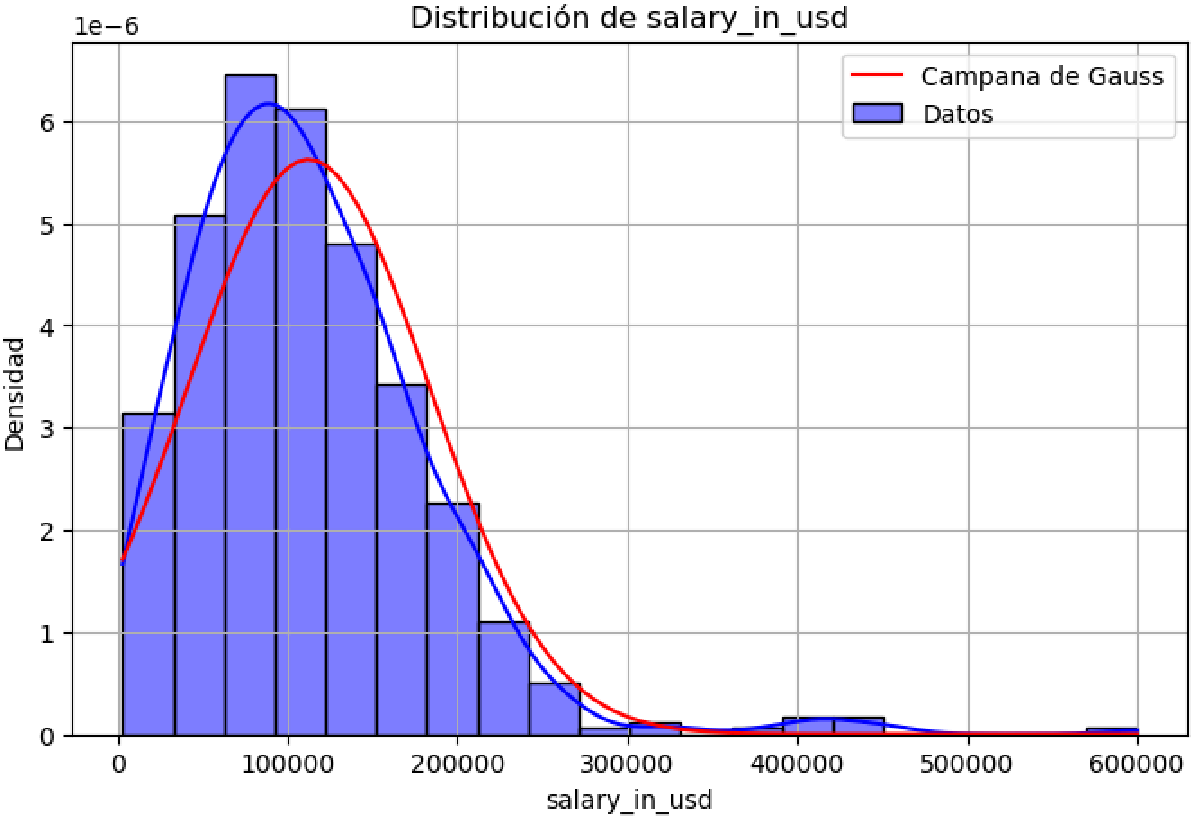
607 rows × 4 columns

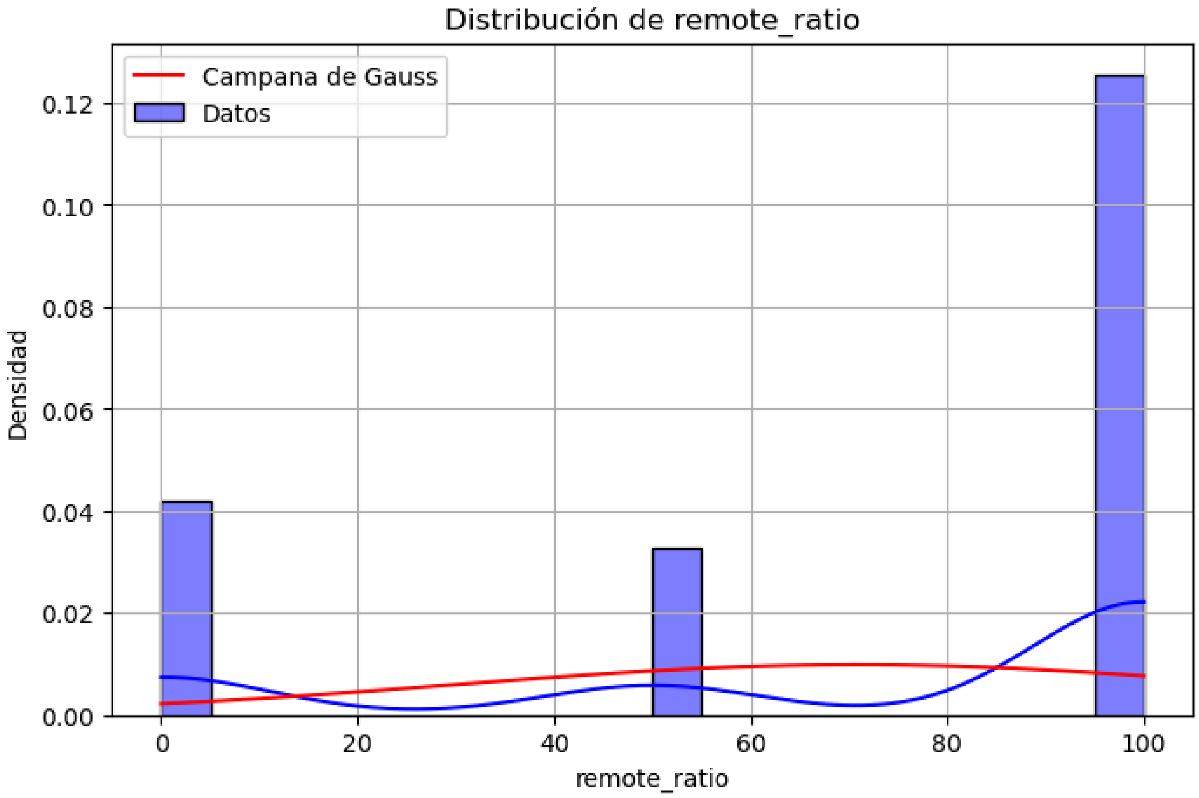
distribución normal de los datos

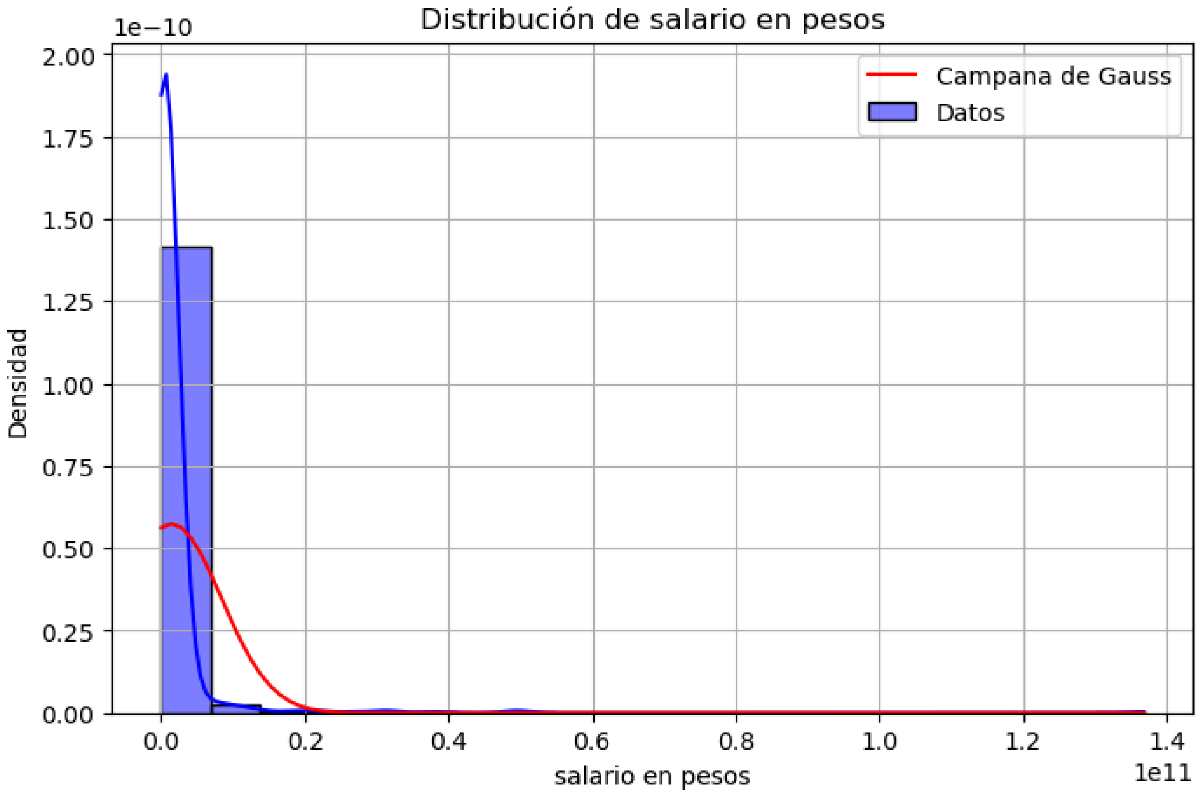
In [170…

```
# Graficar cada variable numérica con su campana de Gauss
for columna in df_cuant.columns:
    plt.figure(figsize=(8, 5))  # Nueva figura para cada variable

    # Histograma con densidad
    sns.histplot(df_cuant[columna], kde=True, bins=20, stat="density", color="blue"

    # Ajuste de la curva normal teórica
    media = df_cuant[columna].mean()
    desviacion = df_cuant[columna].std()
    x = np.linspace(df_cuant[columna].min(), df_cuant[columna].max(), 100) #linea a
    y = norm.pdf(x, media, desviacion)
    plt.plot(x, y, color="red", label="Campana de Gauss")

    # Personalización del gráfico
    plt.title(f"Distribución de {columna}")
    plt.xlabel(columna)
    plt.ylabel("Densidad")
    plt.legend()
    plt.grid(True)

    plt.show()  # Muestra cada gráfico individualmente
```


Distribución de salary
1e−7
Densidad
0
2
4
6
8
Campana de Gauss
Datos
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1e7
salary


Distribución de salary_in_usd
1e−6
Densidad
0
1
2
3
4
5
6
Campana de Gauss
Datos
0
100000
200000
300000
400000
500000
600000
salary_in_usd





la correlacción entre los datos, sirve para revisar la relacion de los datos

```
In [171…  correlacion = df_cuant.corr()
```

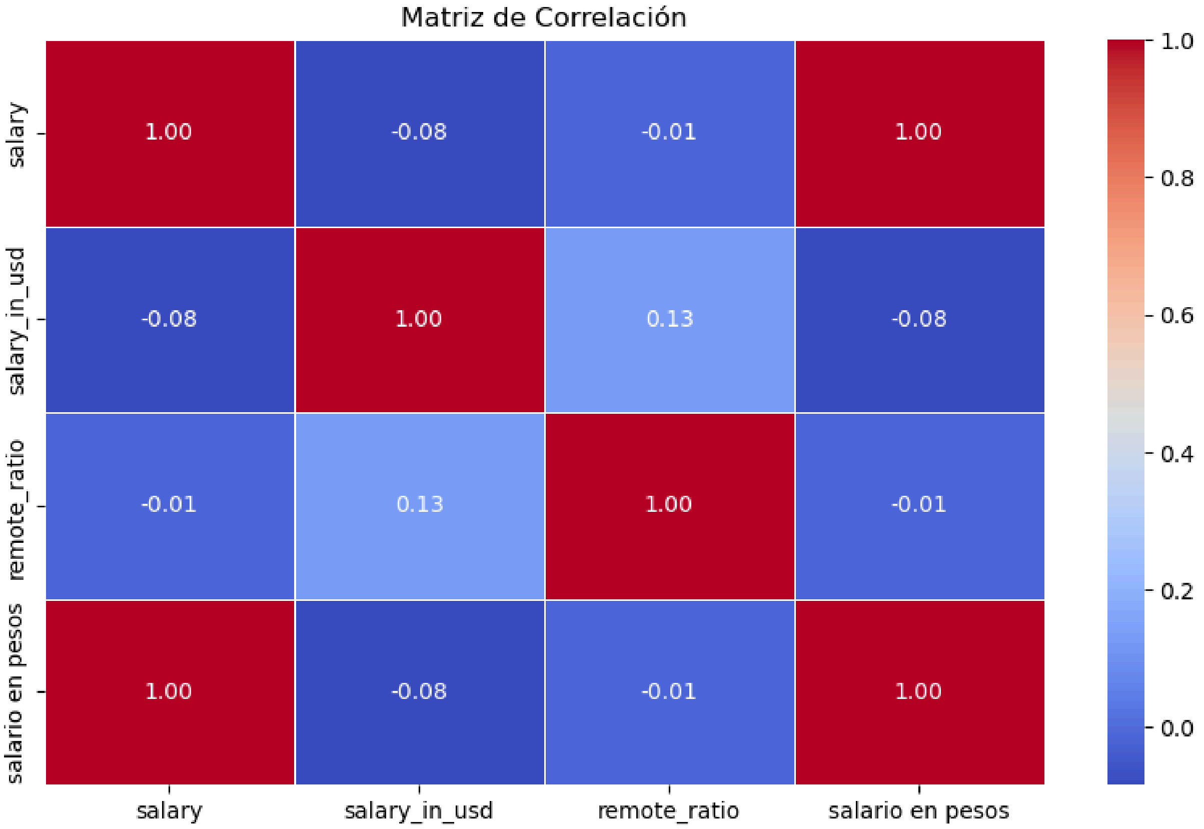
```
In [172…  correlacion
```

Out[172…

	salary	salary_in_usd	remote_ratio	salario en pesos
salary	1.000000	-0.083906	-0.014608	1.000000
salary_in_usd	-0.083906	1.000000	0.132122	-0.083906
remote_ratio	-0.014608	0.132122	1.000000	-0.014608
salario en pesos	1.000000	-0.083906	-0.014608	1.000000

In [173…

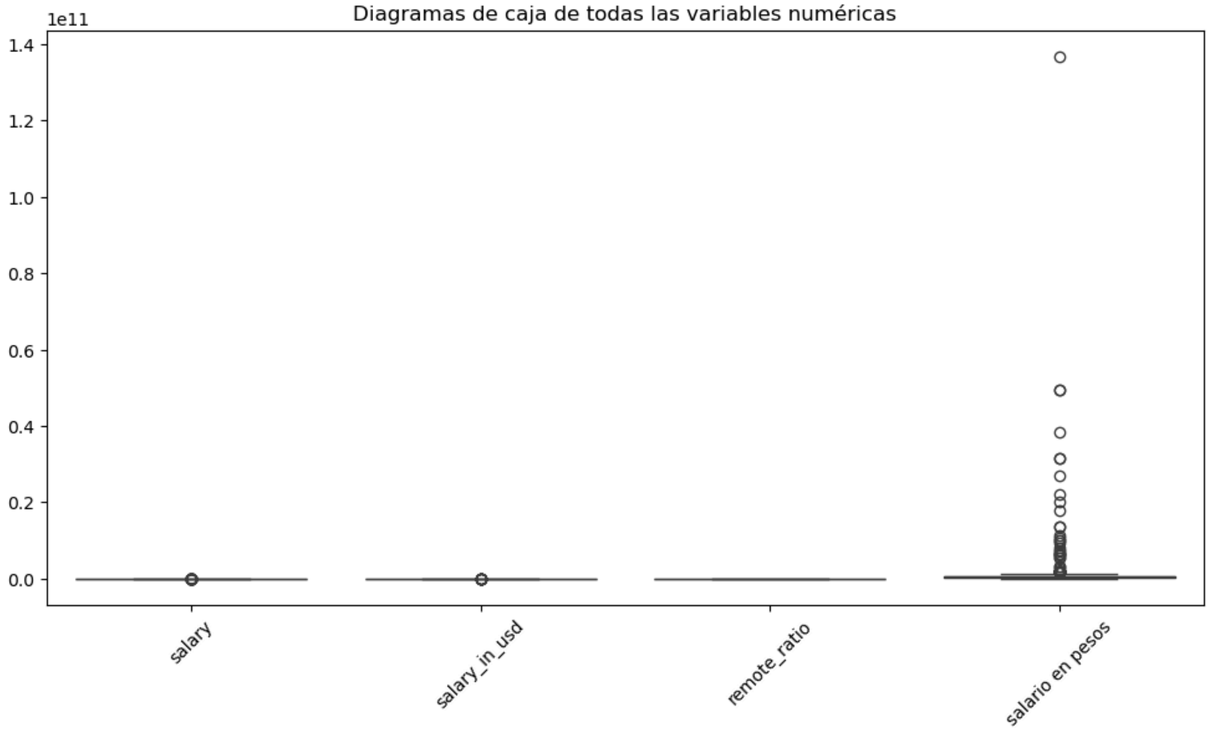
```python
correlacion = df_cuant.corr()
#  🔹 Crear el mapa de calor
plt.figure(figsize=(10, 6))  # Ajustar tamaño de la figura
sns.heatmap(correlacion, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)

#  🔹 Título del gráfico
plt.title("Matriz de Correlación")

#  🔹 Mostrar el gráfico
plt.show()
```



In [174…

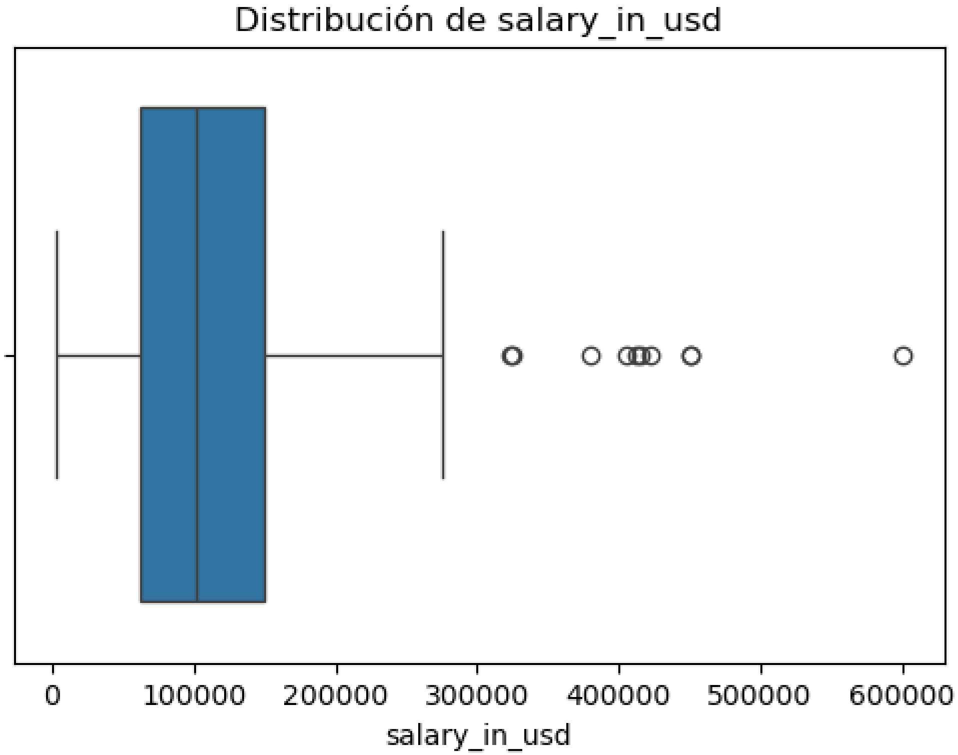
```python
import seaborn as sns
import matplotlib.pyplot as plt

#  🔹 Seleccionar solo las columnas numéricas del DataFrame


#  🔹 Crear un boxplot para todas las columnas numéricas
plt.figure(figsize=(12,6))  # Tamaño del gráfico
sns.boxplot(df_cuant)
```

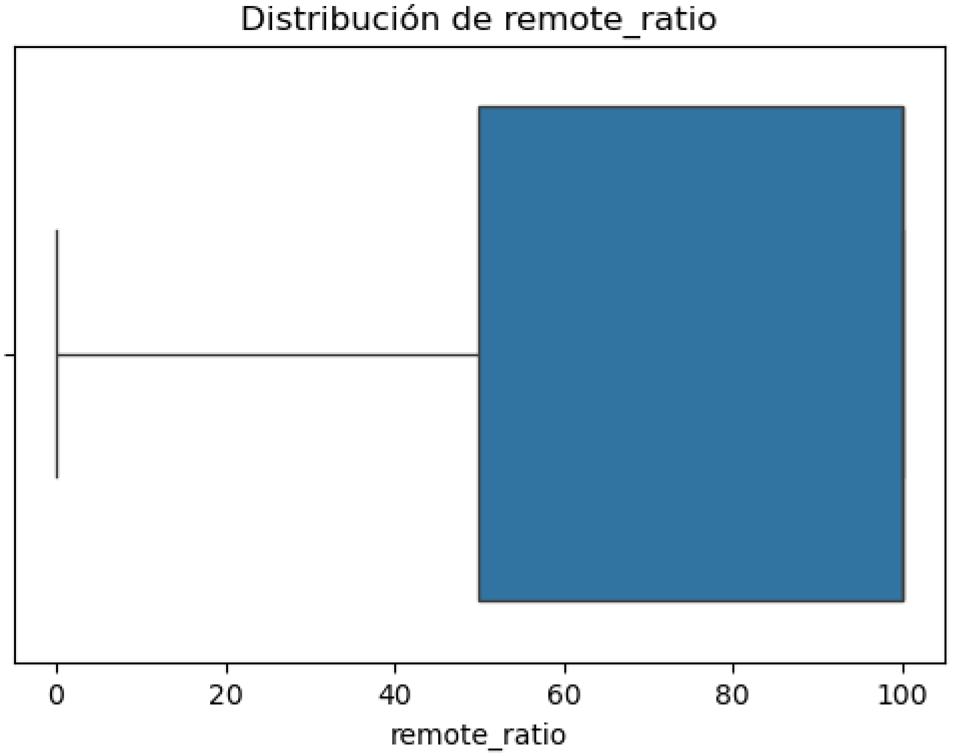
```
# 🔹 Mejorar visualización
plt.xticks(rotation=45)  # Rotar nombres de variables
plt.title("Diagramas de caja de todas las variables numéricas")

# 🔹 Mostrar gráfico
plt.show()
```

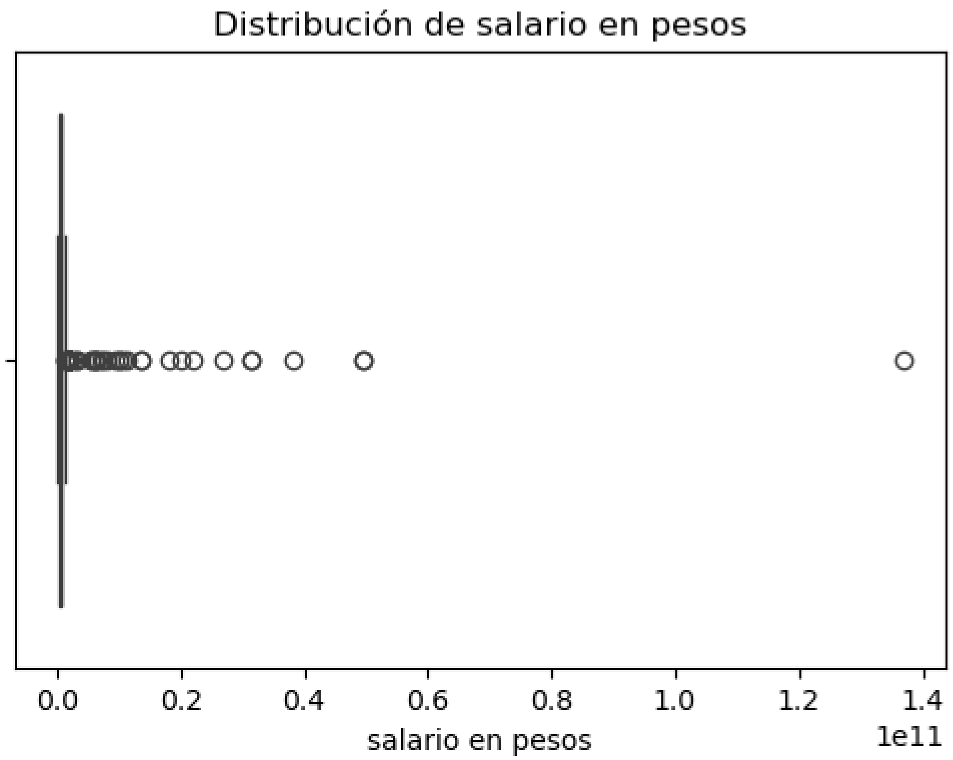


In [175…

```
# 🔹 Recorrer cada columna numérica y hacer un boxplot individual
for i in range(1, df_cuant.shape[1]):
    plt.figure(figsize=(6,4))  # Tamaño de cada gráfico
    sns.boxplot(x=df_cuant.iloc[:, i])
    plt.title(f"Distribución de {df_cuant.columns[i]}")  # Título con el nombre de
    plt.show()
```


Distribución de salary_in_usd
0
100000
200000
300000
400000
500000
600000
salary_in_usd


Distribución de remote_ratio
0
20
40
60
80
100
remote_ratio



In [176…

```
df_cuant.describe()
```

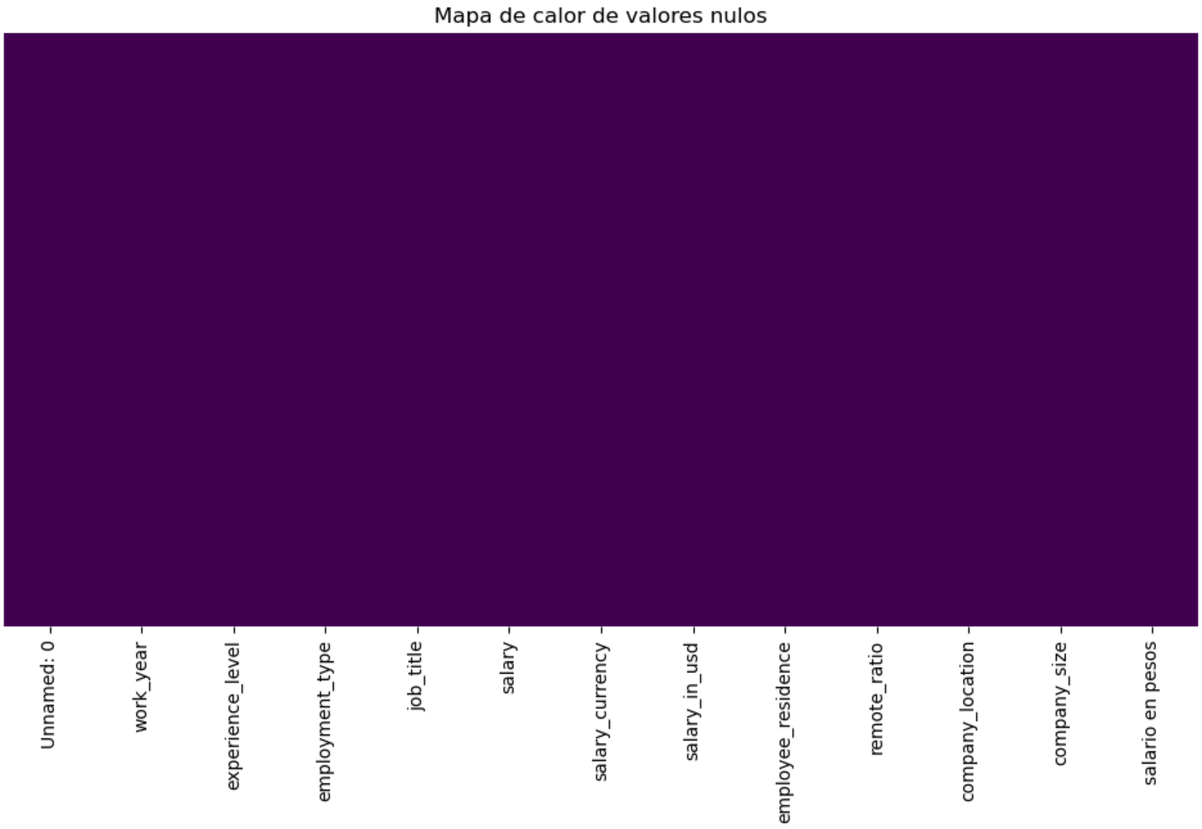
Out[176…

	salary	salary_in_usd	remote_ratio	salario en pesos
count	6.070000e+02	607.000000	607.00000	6.070000e+02
mean	3.240001e+05	112297.869852	70.92257	1.458000e+09
std	1.544357e+06	70957.259411	40.70913	6.949609e+09
min	4.000000e+03	2859.000000	0.00000	1.800000e+07
25%	7.000000e+04	62726.000000	50.00000	3.150000e+08
50%	1.150000e+05	101570.000000	100.00000	5.175000e+08
75%	1.650000e+05	150000.000000	100.00000	7.425000e+08
max	3.040000e+07	600000.000000	100.00000	1.368000e+11

vaores nulos en la data

In [177…

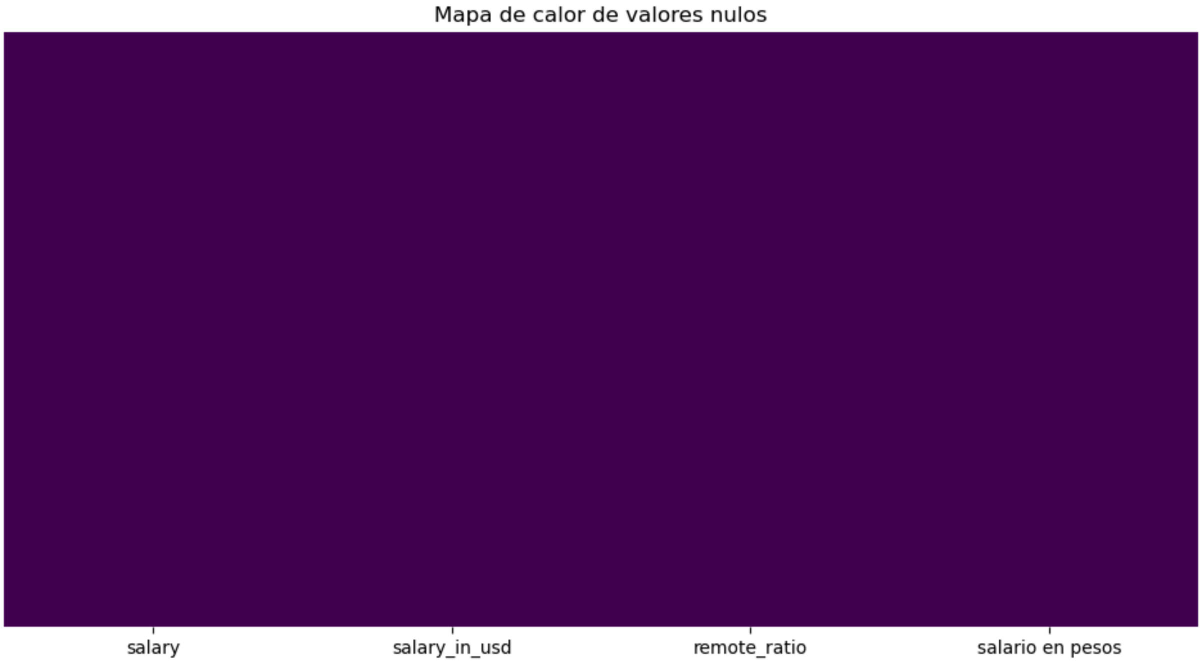
```
plt.figure(figsize=(12,6))
sns.heatmap(df.isnull(), cmap="viridis", cbar=False, yticklabels=False)
plt.title("Mapa de calor de valores nulos")
plt.show()
```



los espacios en blanco son nulos

In [178…

```
plt.figure(figsize=(12,6))
sns.heatmap(df_cuant.isnull(), cmap="viridis", cbar=False, yticklabels=False)
plt.title("Mapa de calor de valores nulos")
plt.show()
```

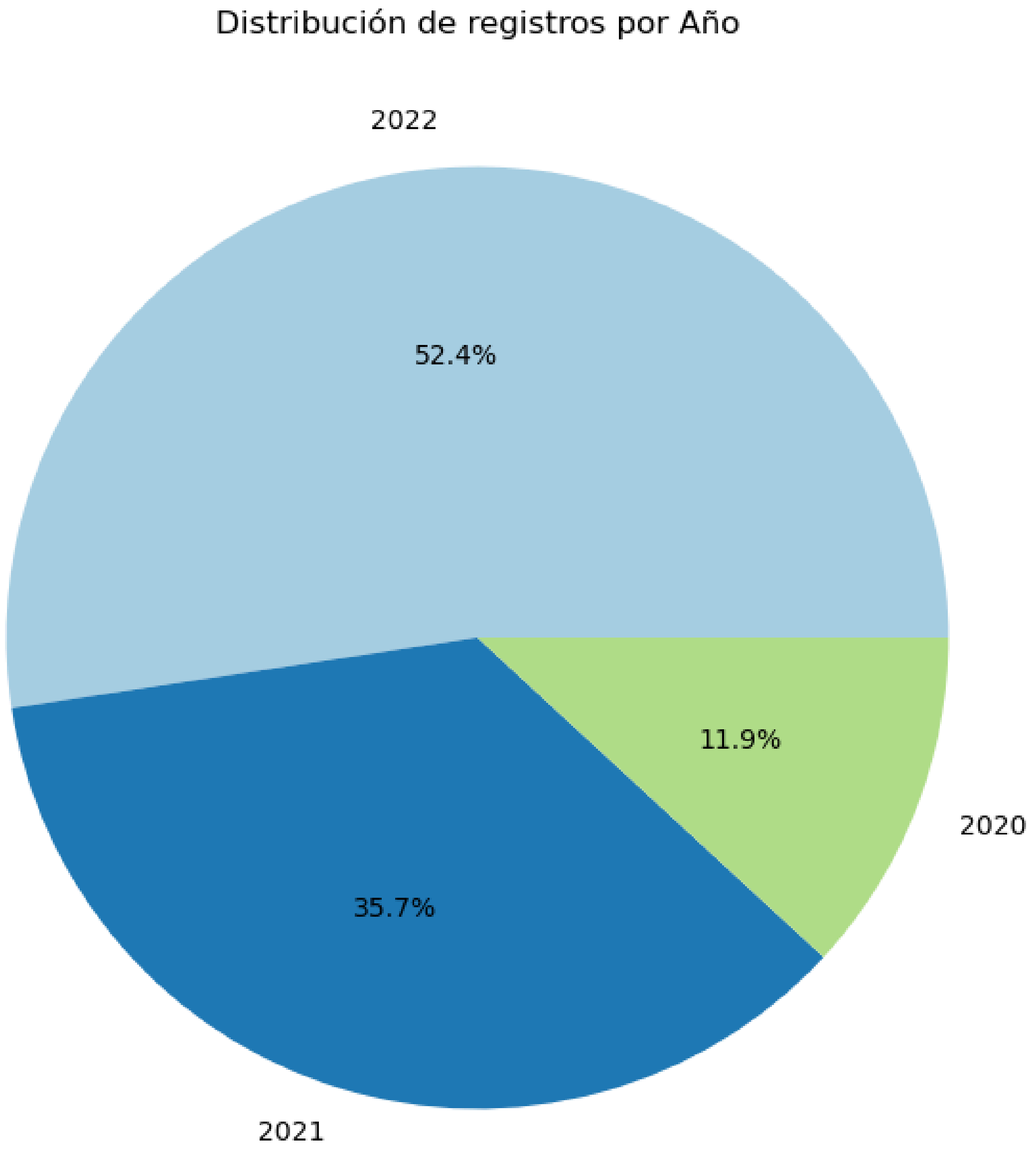


In [179…

```
# Contar cuántos registros hay por año
conteo_años = df["work_year"].value_counts()
print(conteo_años)
```

```
# Crear el gráfico de torta
plt.figure(figsize=(8,8))
plt.pie(conteo_años, labels=conteo_años.index, autopct="%1.1f%%", colors=plt.cm.Pai

# Título y mostrar gráfico
plt.title("Distribución de registros por Año")
plt.show()
```

```
work_year
2022    318
2021    217
2020     72
Name: count, dtype: int64
```



Últimas Exploraciones de la data para aplicar modelos

Ya vimos que si aplicamos una regresión lineal no será el mejor de los resultados, porque analizamos con solo variables cuantitativas, ahora vamos a medir con variables cualitativas de caracter ordinal (la unica que se puede). Entonces tenemos que convertir datos categoricos en números.

In [180…

```
df["experiencia_num"]= df["experience_level"].replace({'EN': 1, 'MI': 2, 'SE': 3, '
df
```

```
C:\Users\darly\AppData\Local\Temp\ipykernel_9088\4027909897.py:1: FutureWarning:

Downcasting behavior in `replace` is deprecated and will be removed in a future vers
ion. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`.
To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting',
True)`
```

Out[180…

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_cu
0	0	2020	MI	FT	Data Scientist	70000	
1	1	2020	SE	FT	Machine Learning Scientist	260000	
2	2	2020	SE	FT	Big Data Engineer	85000	
3	3	2020	MI	FT	Product Data Analyst	20000	
4	4	2020	SE	FT	Machine Learning Engineer	150000	
...	...	...	...	...	...	...	
602	602	2022	SE	FT	Data Engineer	154000	
603	603	2022	SE	FT	Data Engineer	126000	
604	604	2022	SE	FT	Data Analyst	129000	
605	605	2022	SE	FT	Data Analyst	150000	
606	606	2022	MI	FT	AI Scientist	200000	

607 rows × 14 columns

In [182…

```
df["tamanio_campania"]= df["company_size"].replace({'S': 1, 'M': 2, 'L': 3})
df
```

```
C:\Users\darly\AppData\Local\Temp\ipykernel_9088\911041496.py:1: FutureWarning:

Downcasting behavior in `replace` is deprecated and will be removed in a future vers
ion. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`.
To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting',
True)`
```

Out[182…

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_cu
0	0	2020	MI	FT	Data Scientist	70000	
1	1	2020	SE	FT	Machine Learning Scientist	260000	
2	2	2020	SE	FT	Big Data Engineer	85000	
3	3	2020	MI	FT	Product Data Analyst	20000	
4	4	2020	SE	FT	Machine Learning Engineer	150000	
...	...	...	...	...	...	...	
602	602	2022	SE	FT	Data Engineer	154000	
603	603	2022	SE	FT	Data Engineer	126000	
604	604	2022	SE	FT	Data Analyst	129000	
605	605	2022	SE	FT	Data Analyst	150000	
606	606	2022	MI	FT	AI Scientist	200000	

607 rows × 15 columns

In [183…

```python
df.columns
```

Out[183…

```
Index(['Unnamed: 0', 'work_year', 'experience_level', 'employment_type',
       'job_title', 'salary', 'salary_currency', 'salary_in_usd',
       'employee_residence', 'remote_ratio', 'company_location',
       'company_size', 'salario en pesos', 'experiencia_num',
       'tamanio_campania'],
      dtype='object')
```

pronto vamos a almacenar la data que se esta limpiando, entonces eliminamos las columnas que no aportan a la data

In [184…

```
df= df.drop(columns=["Unnamed: 0", "salario en pesos"], inplace=False) #eliminar co
df
```

Out[184…

	work_year	experience_level	employment_type	job_title	salary	salary_currency	sala
0	2020	MI	FT	Data Scientist	70000	EUR	
1	2020	SE	FT	Machine Learning Scientist	260000	USD	
2	2020	SE	FT	Big Data Engineer	85000	GBP	
3	2020	MI	FT	Product Data Analyst	20000	USD	
4	2020	SE	FT	Machine Learning Engineer	150000	USD	
...	...	...	...	...	...	...	
602	2022	SE	FT	Data Engineer	154000	USD	
603	2022	SE	FT	Data Engineer	126000	USD	
604	2022	SE	FT	Data Analyst	129000	USD	
605	2022	SE	FT	Data Analyst	150000	USD	
606	2022	MI	FT	AI Scientist	200000	USD	

607 rows × 13 columns

Vamos creando la data que vamos a medirsacar solo las columnas numericas

In [185…

```
df_analisis= df.select_dtypes(include=['int64', 'float64'])
df_analisis
```

Out[185…

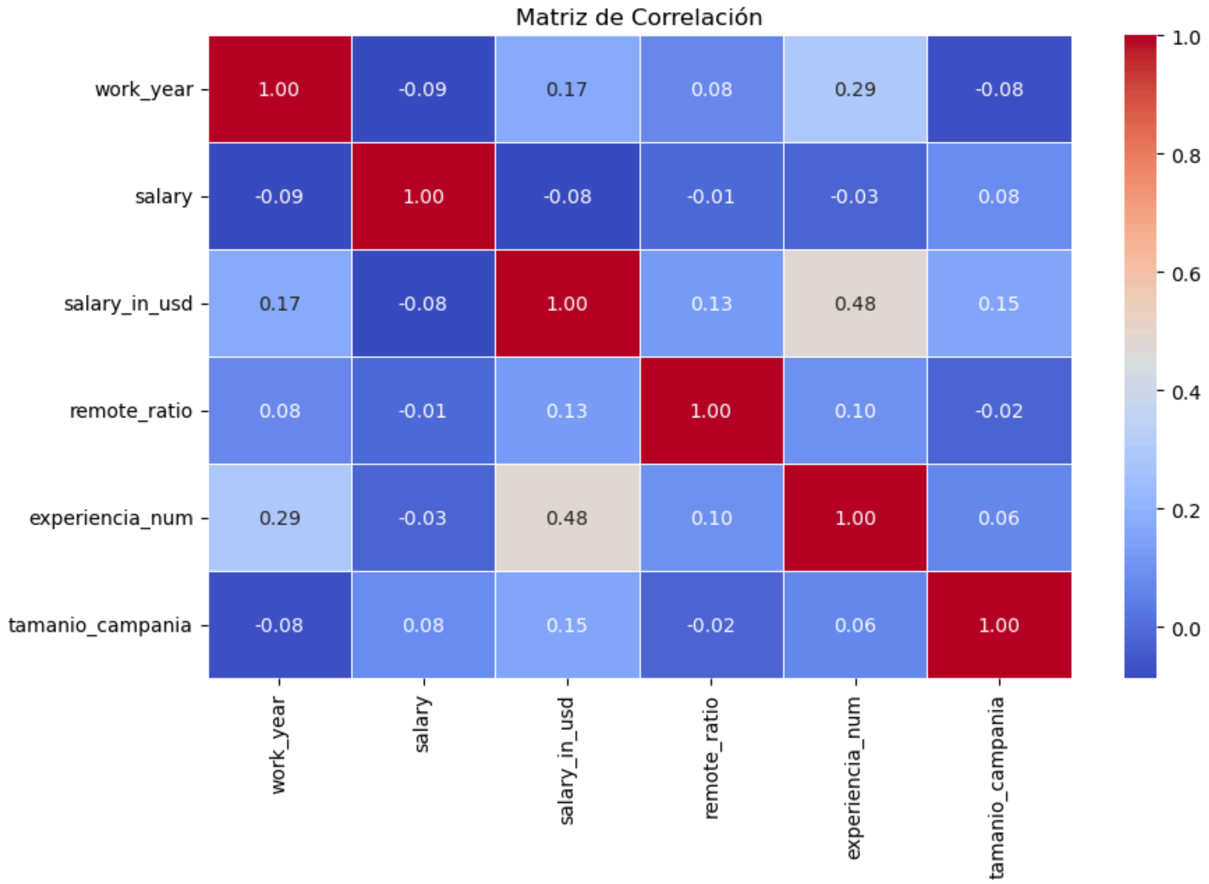
	work_year	salary	salary_in_usd	remote_ratio	experiencia_num	tamanio_campania
0	2020	70000	79833	0	2	3
1	2020	260000	260000	0	3	1
2	2020	85000	109024	50	3	2
3	2020	20000	20000	0	2	1
4	2020	150000	150000	50	3	3
...	...	...	...	...	...	...
602	2022	154000	154000	100	3	2
603	2022	126000	126000	100	3	2
604	2022	129000	129000	0	3	2
605	2022	150000	150000	100	3	2
606	2022	200000	200000	100	2	3

607 rows × 6 columns

análisis de correlación

In [186…

```
correlacion = df_analisis.corr()
# 🔹 Crear el mapa de calor
plt.figure(figsize=(10, 6))  # Ajustar tamaño de la figura
sns.heatmap(correlacion, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)

# 🔹 Título del gráfico
plt.title("Matriz de Correlación")

# 🔹 Mostrar el gráfico
plt.show()
```



Ya vimos algo que cambio... vamos a agregar mas elementos a ver si se modifica Revisemos si asociar los empleos pueden jugar un papel importante

In [187…

```
df['job_title'].nunique() #saca suma de unicos
```

Out[187…

```
50
```

In [249…

```
job_uni= list(df['job_title'].unique()) #una lista de los valores unicos
job_uni
```

Out[249…

```
['Data Scientist',
 'Machine Learning Scientist',
 'Big Data Engineer',
 'Product Data Analyst',
 'Machine Learning Engineer',
 'Data Analyst',
 'Lead Data Scientist',
 'Business Data Analyst',
 'Lead Data Engineer',
 'Lead Data Analyst',
 'Data Engineer',
 'Data Science Consultant',
 'BI Data Analyst',
 'Director of Data Science',
 'Research Scientist',
 'Machine Learning Manager',
 'Data Engineering Manager',
 'Machine Learning Infrastructure Engineer',
 'ML Engineer',
 'AI Scientist',
 'Computer Vision Engineer',
 'Principal Data Scientist',
 'Data Science Manager',
 'Head of Data',
 '3D Computer Vision Researcher',
 'Data Analytics Engineer',
 'Applied Data Scientist',
 'Marketing Data Analyst',
 'Cloud Data Engineer',
 'Financial Data Analyst',
 'Computer Vision Software Engineer',
 'Director of Data Engineering',
 'Data Science Engineer',
 'Principal Data Engineer',
 'Machine Learning Developer',
 'Applied Machine Learning Scientist',
 'Data Analytics Manager',
 'Head of Data Science',
 'Data Specialist',
 'Data Architect',
 'Finance Data Analyst',
 'Principal Data Analyst',
 'Big Data Architect',
 'Staff Data Scientist',
 'Analytics Engineer',
 'ETL Developer',
 'Head of Machine Learning',
 'NLP Engineer',
 'Lead Machine Learning Engineer',
 'Data Analytics Lead']
```

In [189…

```
#un diccionario con los datos unicos y su respectivo valor
dict_job= {}
for i in range(df['job_title'].nunique()):
    dict_job[job_uni[i]]=i+1
```

```
dict_job
```

Out[189…

```
{'Data Scientist': 1,
 'Machine Learning Scientist': 2,
 'Big Data Engineer': 3,
 'Product Data Analyst': 4,
 'Machine Learning Engineer': 5,
 'Data Analyst': 6,
 'Lead Data Scientist': 7,
 'Business Data Analyst': 8,
 'Lead Data Engineer': 9,
 'Lead Data Analyst': 10,
 'Data Engineer': 11,
 'Data Science Consultant': 12,
 'BI Data Analyst': 13,
 'Director of Data Science': 14,
 'Research Scientist': 15,
 'Machine Learning Manager': 16,
 'Data Engineering Manager': 17,
 'Machine Learning Infrastructure Engineer': 18,
 'ML Engineer': 19,
 'AI Scientist': 20,
 'Computer Vision Engineer': 21,
 'Principal Data Scientist': 22,
 'Data Science Manager': 23,
 'Head of Data': 24,
 '3D Computer Vision Researcher': 25,
 'Data Analytics Engineer': 26,
 'Applied Data Scientist': 27,
 'Marketing Data Analyst': 28,
 'Cloud Data Engineer': 29,
 'Financial Data Analyst': 30,
 'Computer Vision Software Engineer': 31,
 'Director of Data Engineering': 32,
 'Data Science Engineer': 33,
 'Principal Data Engineer': 34,
 'Machine Learning Developer': 35,
 'Applied Machine Learning Scientist': 36,
 'Data Analytics Manager': 37,
 'Head of Data Science': 38,
 'Data Specialist': 39,
 'Data Architect': 40,
 'Finance Data Analyst': 41,
 'Principal Data Analyst': 42,
 'Big Data Architect': 43,
 'Staff Data Scientist': 44,
 'Analytics Engineer': 45,
 'ETL Developer': 46,
 'Head of Machine Learning': 47,
 'NLP Engineer': 48,
 'Lead Machine Learning Engineer': 49,
 'Data Analytics Lead': 50}
```

ahora vamos a reemplazar los valores en el df

In [190…

```
df["index_job"]= df['job_title'].replace(dict_job)
df
```

```
C:\Users\darly\AppData\Local\Temp\ipykernel_9088\3371388097.py:2: FutureWarning:

Downcasting behavior in `replace` is deprecated and will be removed in a future vers
ion. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`.
To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting',
True)`
```

Out[190…

	work_year	experience_level	employment_type	job_title	salary	salary_currency	sala
0	2020	MI	FT	Data Scientist	70000	EUR	
1	2020	SE	FT	Machine Learning Scientist	260000	USD	
2	2020	SE	FT	Big Data Engineer	85000	GBP	
3	2020	MI	FT	Product Data Analyst	20000	USD	
4	2020	SE	FT	Machine Learning Engineer	150000	USD	
...	...	...	...	...	...	...	
602	2022	SE	FT	Data Engineer	154000	USD	
603	2022	SE	FT	Data Engineer	126000	USD	
604	2022	SE	FT	Data Analyst	129000	USD	
605	2022	SE	FT	Data Analyst	150000	USD	
606	2022	MI	FT	AI Scientist	200000	USD	

607 rows × 14 columns

In [209…

```
df
```

Out[209…

	work_year	experience_level	employment_type	job_title	salary	salary_currency	sala
0	2020	MI	FT	Data Scientist	70000	EUR	
1	2020	SE	FT	Machine Learning Scientist	260000	USD	
2	2020	SE	FT	Big Data Engineer	85000	GBP	
3	2020	MI	FT	Product Data Analyst	20000	USD	
4	2020	SE	FT	Machine Learning Engineer	150000	USD	
...	...	...	...	...	...	...	
602	2022	SE	FT	Data Engineer	154000	USD	
603	2022	SE	FT	Data Engineer	126000	USD	
604	2022	SE	FT	Data Analyst	129000	USD	
605	2022	SE	FT	Data Analyst	150000	USD	
606	2022	MI	FT	AI Scientist	200000	USD	

607 rows × 14 columns

obtener solo data numerica

In [215…

```
df_analisis = df.select_dtypes(include=['int64', 'float64'])
```

In [216…

```
df_analisis
```

Out[216…

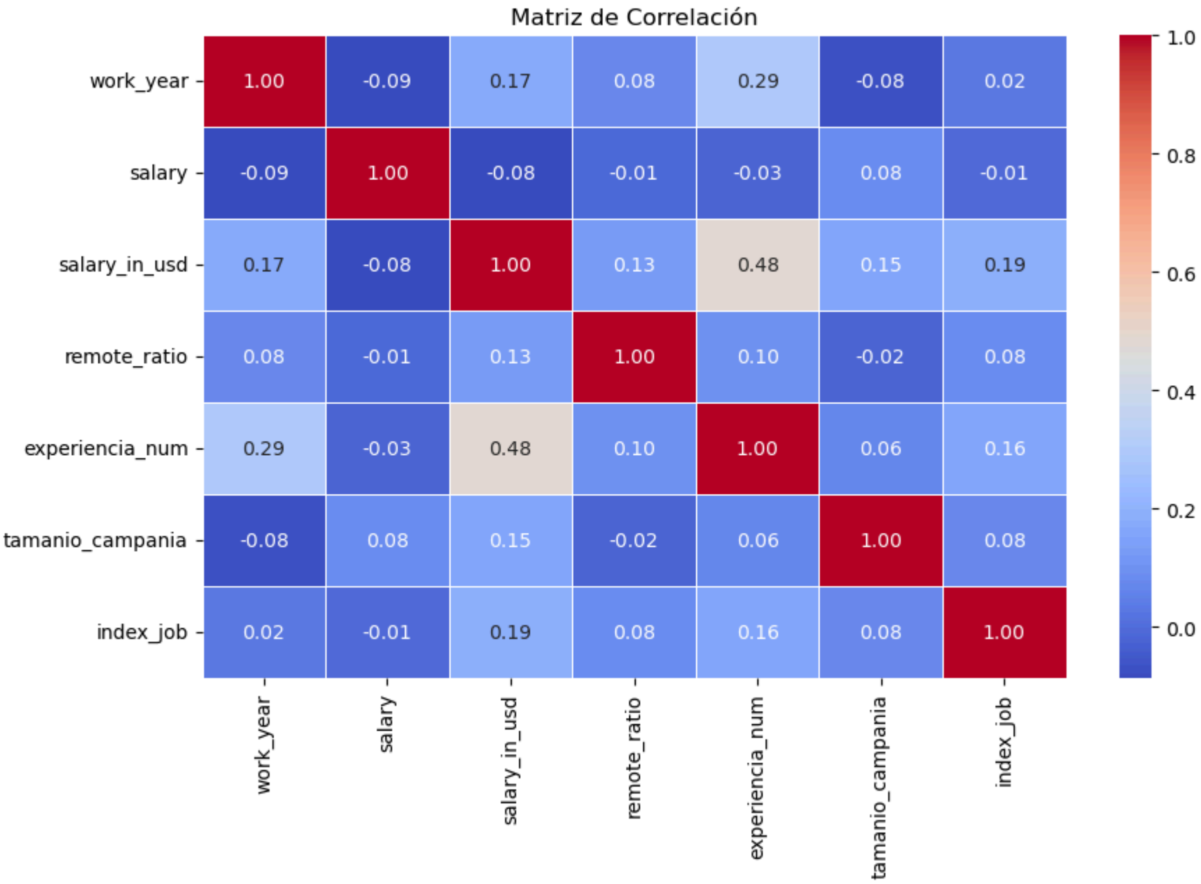
	work_year	salary	salary_in_usd	remote_ratio	experiencia_num	tamanio_campania
0	2020	70000	79833	0	2	3
1	2020	260000	260000	0	3	1
2	2020	85000	109024	50	3	2
3	2020	20000	20000	0	2	1
4	2020	150000	150000	50	3	3
...	...	...	...	...	...	...
602	2022	154000	154000	100	3	2
603	2022	126000	126000	100	3	2
604	2022	129000	129000	0	3	2
605	2022	150000	150000	100	3	2
606	2022	200000	200000	100	2	3

607 rows × 7 columns

de nuevo, análisis de correclación

In [217…

```
correlacion = df_analisis.corr()
# 🔹 Crear el mapa de calor
plt.figure(figsize=(10, 6))  # Ajustar tamaño de la figura
sns.heatmap(correlacion, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)

# 🔹 Título del gráfico
plt.title("Matriz de Correlación")

# 🔹 Mostrar el gráfico
plt.show()
```



NO cambio, toca aplicar oneHOtEncodeng

Guardar las diversas datas

In [219…

```
df.to_csv(r"C:\Users\darly\OneDrive\Escritorio\materialClaseIA\dataSalarios\dataGen
df_analisis.to_csv(r"C:\Users\darly\OneDrive\Escritorio\materialClaseIA\dataSalario
```

Dividir data

Ya ahora nos vamos con la división de las datas para realizar los modelos de predicción y clasificación

In [225…

```
#librerias para modelos de machine learning
from sklearn.model_selection import train_test_split   #divide la data en entrenami
from sklearn.linear_model import LinearRegression      #apicar modelo de regresion
from sklearn.metrics import mean_squared_error, r2_score # metricas del modelo
```

In [228…

```
# Seleccionar la variable independiente (X) y la dependiente (y)
X = df[["experiencia_num"]] # Variable predictora, doble corchete para que retorno
y = df['salary_in_usd']  # Variable objetivo da una serie

```

In [229…

```
X
```

Out[229…

	experiencia_num
0	2
1	3
2	3
3	2
4	3
...	...
602	3
603	3
604	3
605	3
606	2

607 rows × 1 columns

In [231…

```
y
```

Out[231…

```
0       79833
1      260000
2      109024
3       20000
4      150000
        ...  
602    154000
603    126000
604    129000
605    150000
606    200000
Name: salary_in_usd, Length: 607, dtype: int64
```

Partir la data en sets de entranamiento y prueba 80% para entrenar el modelo 20% para evaluar su desempeño

In [233…

```
# se usa para fijar la semilla del generador aleatorio, asegurando que los resultad
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

In [234…

```
X_train, X_test, y_train, y_test
```

```
Out[234…  (     experiencia_num
 9                  3
 227                2
 591                3
 516                3
 132                2
 ..               ...
 71                 2
 106                2
 270                1
 435                2
 102                2

 [485 rows x 1 columns],
      experiencia_num
 563                3
 289                3
 76                 2
 78                 2
 182                2
 ..               ...
 249                3
 365                3
 453                2
 548                3
 235                2

 [122 rows x 1 columns],
 9      125000
 227     88654
 591    144854
 516    152500
 132     38400
         ...
 71      42197
 106    187442
 270     72500
 435     91614
 102     36259
 Name: salary_in_usd, Length: 485, dtype: int64,
 563    140250
 289    135000
 76     100000
 78     270000
 182     26005
         ...
 249    170000
 365    138600
 453    120000
 548     99050
 235    110000
 Name: salary_in_usd, Length: 122, dtype: int64)
```

Aplicar un modelo

Crear y entrenar el modelo de regresión lineal

In [235…

```
# Crear el modelo
modelo = LinearRegression()

# Entrenar el modelo con los datos de entrenamiento
modelo.fit(X_train, y_train)
```

Out[235…

```
▾ LinearRegression
LinearRegression()
```

Cuando entrenamos un modelo de Regresión Lineal con sklearn, el modelo encuentra una ecuación de la recta en la forma:

- $Y = mX + b$

Donde:

- m (pendiente) = Indica cuánto cambia el salario (Y) por cada unidad extra de experiencia (X).
- b (intersección o intercepto) = Es el salario estimado cuando la experiencia es 0.

In [236…

```
# Obtener la pendiente (coeficiente) y la intersección con el eje Y
pendiente =    modelo.coef_[0]
interseccion = modelo.intercept_

print(f"Ecuación de la regresión: Salario = {pendiente:.2f} * Experiencia + {inters
```

```
Ecuación de la regresión: Salario = 44575.96 * Experiencia + 6897.15
```

Los siguientes son los resultados de la predicción

In [237…

```
# Predecir los salarios en el conjunto de prueba, lo
y_pred = modelo.predict(X_test)
```

In [252…

```
y_pred
```

```
Out[252…  array([140625.04456583, 140625.04456583,  96049.08062684,  96049.08062684,
                  96049.08062684,  96049.08062684,  51473.11668785,  51473.11668785,
                  96049.08062684,  51473.11668785,  96049.08062684, 140625.04456583,
                 140625.04456583,  96049.08062684, 140625.04456583, 140625.04456583,
                  96049.08062684,  96049.08062684,  51473.11668785, 140625.04456583,
                 140625.04456583,  96049.08062684, 140625.04456583,  96049.08062684,
                  96049.08062684,  96049.08062684, 140625.04456583,  96049.08062684,
                 140625.04456583,  96049.08062684,  51473.11668785, 140625.04456583,
                 140625.04456583,  51473.11668785,  96049.08062684, 140625.04456583,
                  96049.08062684, 140625.04456583,  51473.11668785, 140625.04456583,
                 140625.04456583, 140625.04456583, 140625.04456583, 140625.04456583,
                 140625.04456583,  96049.08062684,  96049.08062684,  96049.08062684,
                  96049.08062684, 140625.04456583, 140625.04456583,  96049.08062684,
                  96049.08062684, 140625.04456583, 140625.04456583, 140625.04456583,
                 140625.04456583, 185201.00850483,  51473.11668785, 140625.04456583,
                  96049.08062684,  51473.11668785,  96049.08062684, 140625.04456583,
                 140625.04456583,  96049.08062684, 140625.04456583,  96049.08062684,
                 140625.04456583,  96049.08062684, 140625.04456583,  96049.08062684,
                 140625.04456583,  51473.11668785, 140625.04456583, 140625.04456583,
                 140625.04456583, 140625.04456583,  96049.08062684, 140625.04456583,
                 140625.04456583,  51473.11668785,  96049.08062684,  96049.08062684,
                 185201.00850483, 185201.00850483,  51473.11668785, 140625.04456583,
                  96049.08062684, 140625.04456583,  96049.08062684,  96049.08062684,
                  96049.08062684,  96049.08062684, 140625.04456583,  51473.11668785,
                  51473.11668785,  96049.08062684,  96049.08062684, 140625.04456583,
                  96049.08062684, 140625.04456583, 140625.04456583, 185201.00850483,
                 140625.04456583,  51473.11668785, 140625.04456583, 140625.04456583,
                 140625.04456583,  96049.08062684,  96049.08062684,  96049.08062684,
                 140625.04456583, 140625.04456583,  51473.11668785,  96049.08062684,
                  96049.08062684, 140625.04456583, 140625.04456583,  96049.08062684,
                 140625.04456583,  96049.08062684])
```

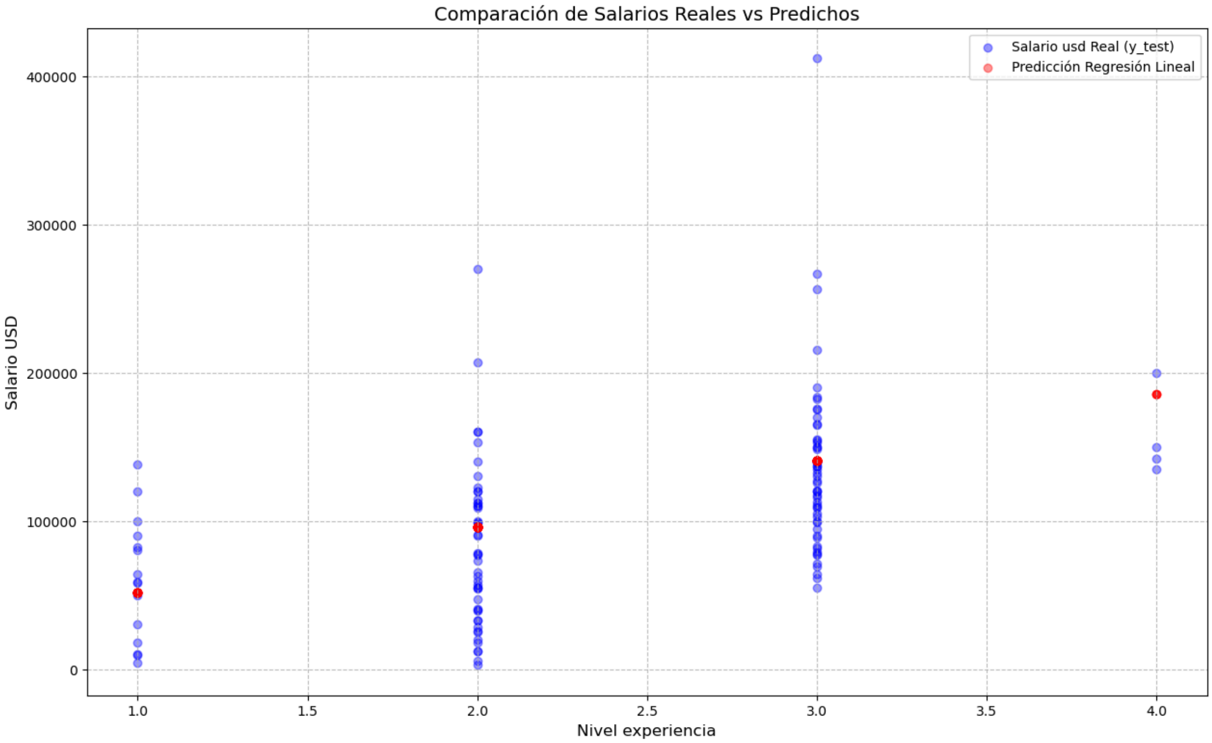
vamos a comparar los resultados

```
In [255…  # Crear un DataFrame con los valores reales y las predicciones
          resultados_predicciones = pd.DataFrame({
              "Salario en USD REAL (y_test)": y_test,
              "Predicción Regresión Lineal ": y_pred,

          })

          # Imprimir los primeros valores en formato de tabla
          print(resultados_predicciones)  # Muestra solo las primeras filas
```

```
     Salario en USD REAL (y_test)  Predicción Regresión Lineal
563                        140250                140625.044566
289                        135000                140625.044566
76                         100000                 96049.080627
78                         270000                 96049.080627
182                         26005                 96049.080627
..                            ...                          ...
249                        170000                140625.044566
365                        138600                140625.044566
453                        120000                 96049.080627
548                         99050                140625.044566
235                        110000                 96049.080627

[122 rows x 2 columns]
```

In [261…

```
# Crear gráfico de dispersión para comparar valores reales y predicciones
plt.figure(figsize=(15,9 ))

# Graficar valores reales (y_test) en azul
plt.scatter(X_test, y_test, color='blue', label="Salario usd Real (y_test)", alpha=

# Graficar predicciones de Regresión Lineal en rojo
plt.scatter(X_test, y_pred, color='red', label="Predicción Regresión Lineal", alpha

# Configurar el gráfico
plt.xlabel("Nivel experiencia", fontsize=12)
plt.ylabel("Salario USD", fontsize=12)
plt.title("Comparación de Salarios Reales vs Predichos", fontsize=14)
plt.legend()
plt.grid(True, linestyle="--", alpha=0.8)

# Mostrar gráfico
plt.show()
```



Calificar un modelo

- El Error Cuadrático Medio (MSE) mide cuánto se alejan las predicciones de los valores reales, calculando el promedio de los errores elevados al cuadrado. Como está en una escala diferente a los datos originales, se recomienda usar la Raíz del MSE (RMSE) para interpretarlo en la misma unidad de la variable objetivo. Un MSE bajo indica mayor precisión del modelo.
- Por otro lado, el Coeficiente de Determinación (R^2) mide qué porcentaje de la variabilidad de los datos es explicado por el modelo, con valores entre 0 y 1 (o negativos si el modelo es muy malo).
- Un R^2 cercano a 1 indica un buen ajuste, mientras que un valor bajo sugiere que el modelo no explica bien los datos.
- Para evaluar si el MSE es grande o pequeño, se debe comparar con la variabilidad de y_test, calculando su rango y desviación estándar.

In [239…

```python
# Calcular el error cuadrático medio (MSE)
mse = mean_squared_error(y_test, y_pred)

# Calcular el coeficiente de determinación R^2
r2 = r2_score(y_test, y_pred)

print(f"Error cuadrático medio (MSE): {mse:.2f}")
print(f"Coeficiente de determinación (R²): {r2:.2f}")
```

```
Error cuadrático medio (MSE): 3019758135.07
Coeficiente de determinación (R²): 0.21
```

Vamos a interpretar estos datos

In [275…

```python
# Calcular el mínimo y máximo de y_test
min_y_test = y_test.min()
max_y_test = y_test.max()

# Calcular el rango de y_test
rango_y_test = max_y_test - min_y_test

# Calcular la desviación estándar de y_test
std_y_test= y_test.std()

#raiz cuadrada de MSE= RMSE
rmse=np.sqrt(mse)


# Mostrar los resultados

print(f"Rango de y_test: {rango_y_test:.2f}")
print(f"Desviación estándar de y_test: {std_y_test:.2f}")
print(f"Raíz de (MSE): (RMSE): {rmse:.2f}")
```

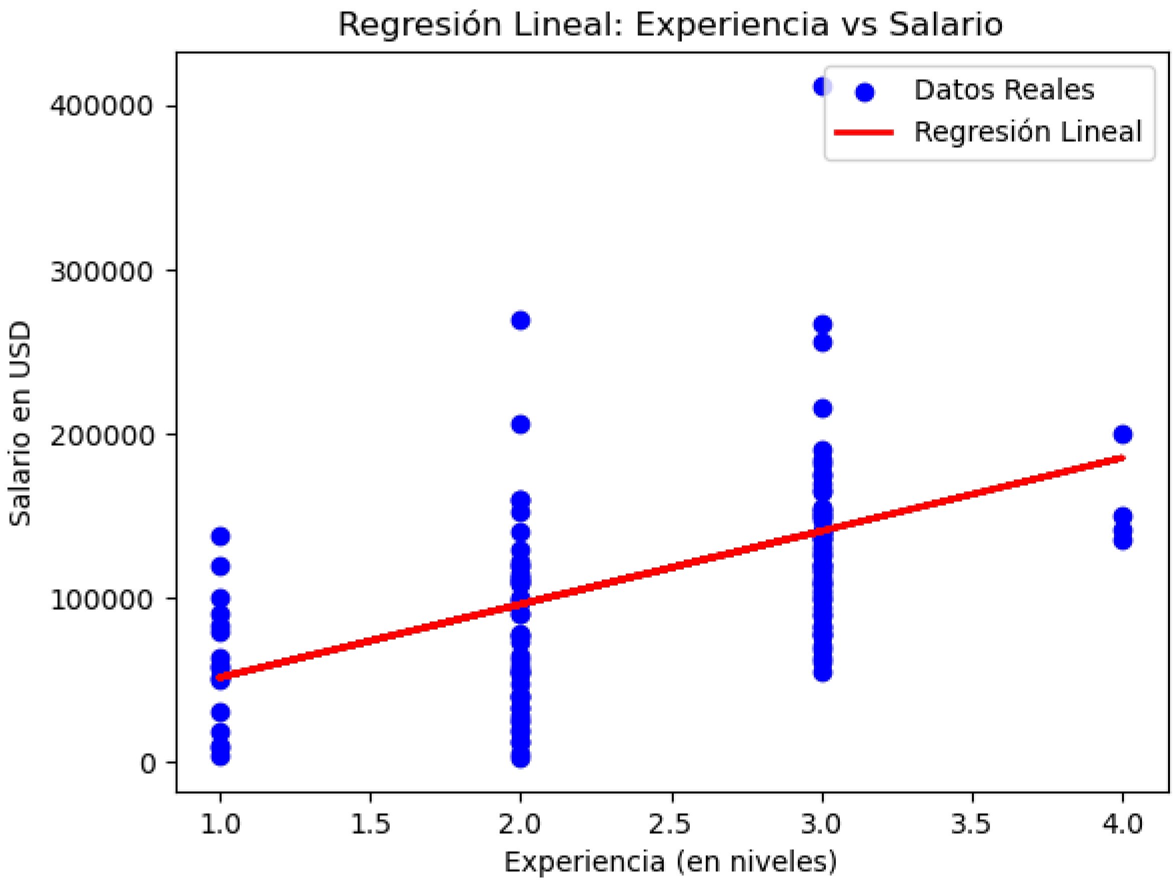
```
Rango de y_test: 409141.00
Desviación estándar de y_test: 62163.04
Raíz de (MSE): (RMSE): 54952.33
```

Interpretación

- El RMSE es menor que la desviación estándar → El modelo tiene cierto nivel de precisión.
- El R^2 es bajo (20%) → El modelo no explica bien los datos, lo que significa que:
- Hay otras variables importantes que no se incluyeron en el modelo.
- El modelo usado (Regresión Lineal) no es el mejor para este problema.
- Los datos pueden contener mucha aleatoriedad o ruido.

In [240…

```python
# Crear gráfico de dispersión
plt.scatter(X_test, y_test, color='blue', label="Datos Reales")
plt.plot(X_test, y_pred, color='red', linewidth=2, label="Regresión Lineal")

# Etiquetas
plt.xlabel("Experiencia (en niveles)")
plt.ylabel("Salario en USD")
plt.title("Regresión Lineal: Experiencia vs Salario")
plt.legend()

# Mostrar gráfico
plt.show()
```


Regresión Lineal: Experiencia vs Salario
Datos Reales
Regresión Lineal
Salario en USD
400000
300000
200000
100000
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Experiencia (en niveles)

```
In [ ]:
```