

Título del trabajo: Reporte

Alumno: Isaac Sánchez Flores

Fecha de entrega: 27/11/2024

Profesor: Jaime Alejandro Romero

Sierra

Materia: Introducción a la ciencia  
de datos



**BUAP**

# **Análisis de productividad**

## **Objetivo**

El objetivo de este proyecto es mejorar la productividad y el rendimiento del personal en un mínimo del 15% en 6 meses.

## **Problema**

La empresa atraviesa un momento crítico en cuanto a productividad y rendimiento, como lo reflejan las evaluaciones recientes que muestran que aproximadamente el 67% del personal está obteniendo un puntaje de rendimiento inferior a 3 en una escala de 5 puntos. Esta tendencia negativa indica la existencia de problemas que están afectando tanto el desempeño como el compromiso del personal. Además, es importante mencionar que la satisfacción promedio de los empleados rondan los 3 puntos en una escala de 5 puntos, lo que podría llevar a renuncias masivas dentro de la empresa, aumentando la carga de trabajo en los demás empleados afectando negativamente la satisfacción y el rendimiento dentro de la empresa.

## **Tecnología y herramientas**

Se estará utilizando el lenguaje de programación Python con las librerías de pandas, matplotlib y numpy para realizar el análisis inicial.

## **Datos**

Para realizar este proyecto, la base de datos que se nos proporcionó nos da información sobre el departamento, genero, edad, puesto de trabajo, años trabajados en la compañía, nivel de educación, puntaje de rendimiento del 1 al 5, salario mensual, horas trabajadas por semana, cantidad de proyectos, horas extras, días de enfermedad tomados, frecuencia de trabajo remoto, tamaño del equipo, horas de capacitación, aumentos/ascensos, satisfacción del empleado y si renunció el empleado o no.

## **Hipótesis iniciales**

Hipótesis 1: El salario mensual es directamente proporcional al rendimiento.

Hipótesis 2: Los empleados con muchos años trabajando en la empresa y con pocos ascensos tienen peor rendimiento.

Hipótesis 3: Las horas de capacitación influyen directamente en el rendimiento.

### **Stakeholders clave**

1. Ejecutivos: Financiamiento y evaluación del proyecto.
2. Departamento de recursos humanos: Gestión del personal e implementación de las políticas.
3. Jefes de departamento: Supervisar que las políticas se sigan.
4. Empleado: Impacto directo en el éxito del proyecto.

### **Preguntas clave**

1. ¿Qué factores afectan el rendimiento?
2. ¿Cuál es el salario promedio que presenta mejor rendimiento?
3. ¿Qué relación hay entre horas de capacitación y rendimiento?
4. ¿Qué tamaño de equipo presenta mejor rendimiento?
5. ¿Qué relación tienen los aumentos con el rendimiento?
6. ¿Cómo influye el nivel educativo en el rendimiento?
7. ¿Cómo se puede aumentar el puntaje de satisfacción?
8. ¿Cómo las horas extras afectan el rendimiento?
9. ¿Cuál es la relación entre edad y rendimiento?
10. ¿Qué departamento presenta peor rendimiento?
11. ¿Cómo influye la antigüedad en la empresa en el rendimiento?
12. ¿Cómo se puede evitar la renuncia o recesión del?

### **Fuentes de datos identificadas**

- Puntaje de satisfacción del empleado.
- Encuesta de rendimiento.
- Medición de horas de capacitación.
- Encuesta sobre el nivel educativo.
- Encuesta de rendimiento por departamento

## **Justificación del proyecto**

El bajo rendimiento y la baja satisfacción de los empleados dentro de la empresa representan un problema crítico que requiere atención inmediata. Esta situación afecta significativamente los ingresos y la calidad de los productos y servicios ofrecidos por la empresa. Un bajo rendimiento entre los empleados suele traducirse en una disminución en la calidad de los productos, lo que impacta negativamente la competitividad frente a otras empresas en el mercado. Si no se toman medidas adecuadas para abordar este problema, la empresa podría enfrentar consecuencias graves, como la pérdida de clientes debido a productos inferiores en comparación con los de la competencia. Además, el problema de la baja satisfacción laboral puede derivar en una mayor rotación de personal, lo que supone otro desafío importante. Las renunciaciones masivas, si no se controla la situación, no solo afectarán la continuidad de las operaciones, sino que también incrementarán los costos de contratación y capacitación de nuevo personal. En resumen, tanto la baja productividad como la insatisfacción de los empleados ponen en riesgo la estabilidad y sostenibilidad de la empresa a largo plazo, haciendo urgente la implementación de estrategias que mejoren el ambiente laboral, motiven a los empleados y eleven su rendimiento, asegurando de esta manera la calidad de nuestros productos y servicios.

## **Cantidad y tipo de datos**

La cantidad de datos que maneja son alrededor de 100,000 por columna, siendo 20 columnas. Cinco columnas son del tipo object, Doce son int64, Dos float64 y una columna es del tipo bool

# Limpieza de datos

## Análisis inicial

### Resumen estadístico de los datos antes de la limpieza:

```
df.describe()
```

#Faltan varias columnas importantes en el resumen estadístico como el puntaje de rendimiento

[124] ✓ 0.0s

	Employee_ID	Years_At_Company	Projects_Handled	Overtime_Hours	Team_Size	Promotions	Employee_Satisfaction_Score
count	121825.000000	121825.000000	121825.000000	121825.000000	121825.000000	121825.000000	119393.000000
mean	50083.834878	4.473918	24.458929	14.508566	10.018247	1.000673	2.998747
std	28889.832419	2.872489	14.469846	8.656905	5.500860	0.815299	1.151397
min	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000
25%	25058.000000	2.000000	12.000000	7.000000	5.000000	0.000000	2.010000
50%	50129.000000	4.000000	24.000000	15.000000	10.000000	1.000000	3.000000
75%	75092.000000	7.000000	37.000000	22.000000	15.000000	2.000000	3.990000
max	100000.000000	10.000000	49.000000	29.000000	19.000000	2.000000	5.000000

+ Code + Markdown

### Tabla que muestra el porcentaje de datos faltantes por columna:

```
# Tabla que muestra el porcentaje de datos faltantes en cada columna
porcentaje_datos_faltantes = (df.isnull().mean() * 100).round(2).astype(str) + '%'
print(porcentaje_datos_faltantes)
```

#En total alrededor del 80% de datos es nulo

[127] ✓ 0.0s Python

Employee_ID	4.0%
Department	4.0%
Gender	4.0%
Age	4.0%
Job_Title	4.0%
Hire_Date	4.0%
Years_At_Company	4.0%
Education_Level	4.0%
Performance_Score	4.0%
Monthly_Salary	4.0%
Work_Hours_Per_Week	4.0%
Projects_Handled	4.0%
Overtime_Hours	4.0%
Sick_Days	4.0%
Remote_Work_Frequency	4.0%
Team_Size	4.0%
Training_Hours	4.0%
Promotions	4.0%
Employee_Satisfaction_Score	5.92%
Resigned	4.0%
dtype: object	

Total de datos repetidos encontrados: (Solo en Employee\_ID ya que en los demás es irrelevante)

```
df.duplicated('Employee_ID').sum() #Comprobamos cuantos datos duplicados hay en Employee_ID

#Tenemos 30032 duplicados
```

✓ 0.0s

30032

Descripción de los tipos de datos originales y los problemas encontrados:

```
df.info()
```

```
#La mayoría de datos está en formato object, tengo que cambiar a float o int según sea conveniente
```

✓ 0.0s

Python

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 126901 entries, 0 to 126900
```

```
Data columns (total 20 columns):
```

#	Column	Non-Null Count	Dtype
0	Employee_ID	121825 non-null	float64
1	Department	121825 non-null	object
2	Gender	121825 non-null	object
3	Age	121825 non-null	object
4	Job_Title	121825 non-null	object
5	Hire_Date	121825 non-null	object
6	Years_At_Company	121825 non-null	float64
7	Education_Level	121825 non-null	object
8	Performance_Score	121825 non-null	object
9	Monthly_Salary	121825 non-null	object
10	Work_Hours_Per_Week	121825 non-null	object
11	Projects_Handled	121825 non-null	float64
12	Overtime_Hours	121825 non-null	float64
13	Sick_Days	121825 non-null	object
14	Remote_Work_Frequency	121825 non-null	object
15	Team_Size	121825 non-null	float64
16	Training_Hours	121825 non-null	object
17	Promotions	121825 non-null	float64
18	Employee_Satisfaction_Score	119393 non-null	float64
19	Resigned	121825 non-null	object

```
dtypes: float64(7), object(13)
```

```
memory usage: 19.4+ MB
```

Los datos originalmente eran de tipo numérico, en su mayoría, pero ahora todos los datos estaban en formato object, esto provoca que datos importantes como el Performance\_Score no aparezcan en el resumen estadístico.

Otro problema que encontré a la hora de hacer el análisis inicial de la base de datos fue que gran parte de las columnas contenían valores inválidos tales como “bbb”

```
lista_columnas = df.columns

for c in lista_columnas:
    print(f'En la columna {c} los bbb son: {df[df[c] == 'bbb'].shape[0]}')

#Revisamos cuantos bbb hay en cada columna
```

[130] ✓ 0.0s Python

... En la columna Employee\_ID los bbb son: 0  
En la columna Department los bbb son: 2430  
En la columna Gender los bbb son: 0  
En la columna Age los bbb son: 2436  
En la columna Job\_Title los bbb son: 0  
En la columna Hire\_Date los bbb son: 0  
En la columna Years\_At\_Company los bbb son: 0  
En la columna Education\_Level los bbb son: 2445  
En la columna Performance\_Score los bbb son: 2434  
En la columna Monthly\_Salary los bbb son: 2439  
En la columna Work\_Hours\_Per\_Week los bbb son: 2425  
En la columna Projects\_Handled los bbb son: 0  
En la columna Overtime\_Hours los bbb son: 0  
En la columna Sick\_Days los bbb son: 2434  
En la columna Remote\_Work\_Frequency los bbb son: 2441  
En la columna Team\_Size los bbb son: 0  
En la columna Training\_Hours los bbb son: 2445  
En la columna Promotions los bbb son: 0  
En la columna Employee\_Satisfaction\_Score los bbb son: 0  
En la columna Resigned los bbb son: 0

## Proceso de limpieza

Para la limpieza de la base de datos además de la librería pandas utilicé la librería de numpy por razones que explicaré a continuación. Las funciones de pandas que utilicé fueron las de eliminación de duplicados y valores NaN, conversión de datos a un formato en específico y llenado de valores NaN ya sea con una leyenda que diga ‘Sin Datos’ o con el promedio de cada columna. La única función que utilicé de numpy fue la que permite convertir datos a NaN, esto lo hice para sustituir la cadena ‘bbb’ por valores que me fueran más convenientes con ayuda de la función de pandas mencionada anteriormente.

### Antes de eliminar duplicados:

```
df.duplicated('Employee_ID').sum() #Comprobamos cuantos datos duplicados hay en Employee_ID

#Tenemos 30032 duplicados
```

[128] ✓ 0.0s

... 30032

## Después de eliminar duplicados:

```
#Primero eliminamos los duplicados en Employee_ID

df = df.drop_duplicates(subset=['Employee_ID'])

df = df.reset_index(drop=True)

df.duplicated(subset=['Employee_ID']).sum()
```

[131] ✓ 0.0s

Python

## Antes de transformar los datos a numérico:

```
df.info()
```

```
#La mayoría de datos está en formato object, tengo que cambiar a float o int según sea conveniente
```

[150] ✓ 0.0s

Python

```
... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 126901 entries, 0 to 126900
Data columns (total 20 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Employee_ID           121825 non-null float64
1   Department            121825 non-null object
2   Gender                121825 non-null object
3   Age                  121825 non-null object
4   Job_Title             121825 non-null object
5   Hire_Date             121825 non-null object
6   Years_At_Company      121825 non-null float64
7   Education_Level       121825 non-null object
8   Performance_Score     121825 non-null object
9   Monthly_Salary        121825 non-null object
10  Work_Hours_Per_Week    121825 non-null object
11  Projects_Handled       121825 non-null float64
12  Overtime_Hours         121825 non-null float64
13  Sick_Days             121825 non-null object
14  Remote_Work_Frequency 121825 non-null object
15  Team_Size             121825 non-null float64
16  Training_Hours         121825 non-null object
17  Promotions            121825 non-null float64
18  Employee_Satisfaction_Score 119393 non-null float64
19  Resigned              121825 non-null object
dtypes: float64(7), object(13)
memory usage: 19.4+ MB
```



Después de transformar los datos a numérico:

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 96869 entries, 0 to 96868
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   Employee_ID           96868 non-null  float64
 1   Department            92987 non-null  object
 2   Gender                92979 non-null  object
 3   Age                  91058 non-null  float64
 4   Job_Title             92971 non-null  object
 5   Hire_Date             92983 non-null  object
 6   Years_At_Company      92990 non-null  float64
 7   Education_Level       92999 non-null  object
 8   Performance_Score     91186 non-null  float64
 9   Monthly_Salary        91053 non-null  float64
10   Work_Hours_Per_Week   91142 non-null  float64
11   Projects_Handled      93005 non-null  float64
12   Overtime_Hours        93028 non-null  float64
13   Sick_Days             91133 non-null  float64
14   Remote_Work_Frequency 91200 non-null  float64
15   Team_Size             92950 non-null  float64
16   Training_Hours        91104 non-null  float64
17   Promotions            93009 non-null  float64
18   Employee_Satisfaction_Score 91204 non-null float64
19   Resigned              92977 non-null  object
dtypes: float64(14), object(6)
memory usage: 14.8+ MB
```

Antes de convertir 'bbb' a NaN:

df									
	Employee_ID	Department	Gender	Age	Job_Title	Hire_Date	Years_At_Company	Education_Level	Performance_Score
0	1.0	IT	Male	55	Specialist	NaN	2.0	High School	91186.0
1	2.0	Finance	Male	29	Developer	2024-04-18 08:03:05.556036	0.0	High School	91186.0
2	3.0	Finance	Male	NaN	Specialist	2015-10-26 08:03:05.556036	8.0	High School	91186.0
3	4.0	bbb	Female	48	Analyst	2016-10-22 08:03:05.556036	7.0	Bachelor	91186.0
4	5.0	Engineering	Female	36	NaN	2021-07-23 08:03:05.556036	3.0	Bachelor	91186.0
...	...	...	...	...	...	...	...	...	...
126896	9646.0	Finance	Male	38	Engineer	2021-06-07 08:03:05.556036	3.0	Bachelor	91186.0
126897	9646.0	Finance	Male	38	Engineer	2023-09-05 08:03:05.556036	3.0	Bachelor	91186.0

Después de convertir ‘bbb’ a NaN:

[159]

```
#Convertimos los bbb a NaN utilizando la librería de Numpy para luego llenar con el promedio o 'Sin dato', según sea conveniente

df.replace('bbb' , np.nan, inplace=True)

df
```

✓ 0.0s

...

	Employee_ID	Department	Gender	Age	Job_Title	Hire_Date	Years_At_Company	Education_Level	Performance_Score	Monthly_Sala
0	1.0	IT	Male	55.0	Specialist	NaN	2.0	High School	5.0	6750
1	2.0	Finance	Male	29.0	Developer	2024-04-18 08:03:05.556036	0.0	High School	5.0	7500
2	3.0	Finance	Male	NaN	Specialist	2015-10-26 08:03:05.556036	8.0	High School	3.0	Na
3	4.0	NaN	Female	48.0	Analyst	2016-10-22 08:03:05.556036	7.0	Bachelor	2.0	4800
4	5.0	Engineering	Female	36.0	NaN	2021-07-23 08:03:05.556036	3.0	Bachelor	2.0	4800
...	...	...	...	...	...	...	...	...	...	...
96864	72705.0	Engineering	Female	27.0	Specialist	2019-07-23 08:03:05.556036	NaN	Bachelor	NaN	5400
96865	3248.0	Operations	Male	35.0	Analyst	2017-08-29 08:03:05.556036	NaN	Bachelor	2.0	4800

Antes de remplazar NaN por ‘Sin Dato’ o el promedio:

[159]

```
#Convertimos los bbb a NaN utilizando la librería de Numpy para luego llenar con el promedio o 'Sin dato', según sea conveniente

df.replace('bbb' , np.nan, inplace=True)

df
```

✓ 0.0s

...

	Employee_ID	Department	Gender	Age	Job_Title	Hire_Date	Years_At_Company	Education_Level	Performance_Score	Monthly_Sala
0	1.0	IT	Male	55.0	Specialist	NaN	2.0	High School	5.0	6750
1	2.0	Finance	Male	29.0	Developer	2024-04-18 08:03:05.556036	0.0	High School	5.0	7500
2	3.0	Finance	Male	NaN	Specialist	2015-10-26 08:03:05.556036	8.0	High School	3.0	Na
3	4.0	NaN	Female	48.0	Analyst	2016-10-22 08:03:05.556036	7.0	Bachelor	2.0	4800
4	5.0	Engineering	Female	36.0	NaN	2021-07-23 08:03:05.556036	3.0	Bachelor	2.0	4800
...	...	...	...	...	...	...	...	...	...	...
96864	72705.0	Engineering	Female	27.0	Specialist	2019-07-23 08:03:05.556036	NaN	Bachelor	NaN	5400
96865	3248.0	Operations	Male	35.0	Analyst	2017-08-29 08:03:05.556036	NaN	Bachelor	2.0	4800

Después de reemplazar 'bbb' por 'Sin Dato' o el promedio:

...	Employee_ID	Department	Gender	Age	Job_Title	Hire_Date	Years_At_Company	Education_Level	Performance_Score	Monthly_Salary	V
	0	1.0	IT	Male	55.000000	Specialist	NaN	2.000000	High School	5.000000	6750.000000
	1	2.0	Finance	Male	29.000000	Developer	2024-04-18 08:03:05.556036	0.000000	High School	5.000000	7500.000000
	2	3.0	Finance	Male	41.024325	Specialist	2015-10-26 08:03:05.556036	8.000000	High School	3.000000	6404.100908
	3	4.0	Sin dato	Female	48.000000	Analyst	2016-10-22 08:03:05.556036	7.000000	Bachelor	2.000000	4800.000000
	4	5.0	Engineering	Female	36.000000	Sin dato	2021-07-23 08:03:05.556036	3.000000	Bachelor	2.000000	4800.000000
...	...	...	...	...	...	...	...	...	...	...	...
	96864	72705.0	Engineering	Female	27.000000	Specialist	2019-07-23 08:03:05.556036	4.478632	Bachelor	2.996184	5400.000000
	96865	3248.0	Operations	Male	35.000000	Analyst	2017-08-29 08:03:05.556036	4.478632	Bachelor	2.000000	4800.000000
	96866	66901.0	Sales	Male	41.024325	Specialist	2015-02-01 08:03:05.556036	9.000000	High School	4.000000	6300.000000
	96867	21770.0	Marketing	Male	33.000000	Technician	2016-10-22 08:03:05.556036	7.000000	Bachelor	3.000000	4550.000000
	96868	33177.0	Operations	Other	39.000000	Specialist	2020-12-24 08:03:05.556036	3.000000	Bachelor	3.000000	5850.000000

Antes de aplicar dropna en Employee\_ID y Hire\_Date:

```
df.info()
[212] ✓ 0.0s

... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 96869 entries, 0 to 96868
Data columns (total 20 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Employee_ID                          96869 non-null  object
1   Department                           96869 non-null  object
2   Gender                               96869 non-null  object
3   Age                                  96869 non-null  float64
4   Job_Title                            96869 non-null  object
5   Hire_Date                            92983 non-null  object
6   Years_At_Company                     96869 non-null  float64
7   Education_Level                      96869 non-null  object
8   Performance_Score                   96869 non-null  float64
9   Monthly_Salary                      96869 non-null  float64
10  Work_Hours_Per_Week                 96869 non-null  float64
11  Projects_Handled                    96869 non-null  float64
12  Overtime_Hours                     96869 non-null  float64
13  Sick_Days                          96869 non-null  float64
14  Remote_Work_Frequency               96869 non-null  float64
15  Team_Size                           96869 non-null  float64
16  Training_Hours                      96869 non-null  float64
17  Promotions                          96869 non-null  float64
18  Employee_Satisfaction_Score         96869 non-null  float64
19  Resigned                           96869 non-null  object
dtypes: float64(13), object(7)
memory usage: 14.8+ MB
```

Después de aplicar dropna en Employee\_ID y Hire\_Date:

```
> ✓ #Verificamos que no haya valores nulos
df.info()
266] ✓ 0.0s
... <class 'pandas.core.frame.DataFrame'>
Index: 92982 entries, 1 to 96868
Data columns (total 20 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Employee_ID                          92982 non-null  float64
1   Department                           92982 non-null  object
2   Gender                               92982 non-null  object
3   Age                                   92982 non-null  float64
4   Job Title                            92982 non-null  object
5   Hire_Date                            92982 non-null  object
6   Years_At_Company                     92982 non-null  float64
7   Education_Level                      92982 non-null  object
8   Performance_Score                    92982 non-null  float64
9   Monthly_Salary                       92982 non-null  float64
10  Work_Hours_Per_Week                  92982 non-null  float64
11  Projects_Handled                     92982 non-null  float64
12  Overtime_Hours                       92982 non-null  float64
13  Sick_Days                            92982 non-null  float64
14  Remote_Work_Frequency                92982 non-null  float64
15  Team_Size                            92982 non-null  float64
16  Training_Hours                       92982 non-null  float64
17  Promotions                           92982 non-null  float64
18  Employee_Satisfaction_Score          92982 non-null  float64
19  Resigned                             92982 non-null  object
dtypes: float64(14), object(6)
memory usage: 14.9+ MB
```

Antes de convertir todos los datos a su respectivo formato:

```
> ✓ #Verificamos que no haya valores nulos
df.info()
266] ✓ 0.0s
... <class 'pandas.core.frame.DataFrame'>
Index: 92982 entries, 1 to 96868
Data columns (total 20 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Employee_ID                          92982 non-null  float64
1   Department                           92982 non-null  object
2   Gender                               92982 non-null  object
3   Age                                   92982 non-null  float64
4   Job Title                            92982 non-null  object
5   Hire_Date                            92982 non-null  object
6   Years_At_Company                     92982 non-null  float64
7   Education_Level                      92982 non-null  object
8   Performance_Score                    92982 non-null  float64
9   Monthly_Salary                       92982 non-null  float64
10  Work_Hours_Per_Week                  92982 non-null  float64
11  Projects_Handled                     92982 non-null  float64
12  Overtime_Hours                       92982 non-null  float64
13  Sick_Days                            92982 non-null  float64
14  Remote_Work_Frequency                92982 non-null  float64
15  Team_Size                            92982 non-null  float64
16  Training_Hours                       92982 non-null  float64
17  Promotions                           92982 non-null  float64
18  Employee_Satisfaction_Score          92982 non-null  float64
19  Resigned                             92982 non-null  object
dtypes: float64(14), object(6)
memory usage: 14.9+ MB
```

Después de convertir todos los datos a su respectivo formato:

```
df.info()
[272] ✓ 0.0s

... <class 'pandas.core.frame.DataFrame'>
Index: 92982 entries, 1 to 96868
Data columns (total 20 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Employee_ID                          92982 non-null  float64
1   Department                           92982 non-null  object
2   Gender                               92982 non-null  object
3   Age                                  92982 non-null  int64
4   Job Title                            92982 non-null  object
5   Hire Date                            92982 non-null  datetime64[ns]
6   Years_At_Company                     92982 non-null  int64
7   Education_Level                      92982 non-null  object
8   Performance_Score                   92982 non-null  int64
9   Monthly_Salary                      92982 non-null  int64
10  Work_Hours_Per_Week                 92982 non-null  int64
11  Projects_Handled                    92982 non-null  int64
12  Overtime_Hours                     92982 non-null  int64
13  Sick_Days                          92982 non-null  int64
14  Remote_Work_Frequency               92982 non-null  int64
15  Team_Size                           92982 non-null  int64
16  Training_Hours                     92982 non-null  int64
17  Promotions                          92982 non-null  int64
18  Employee_Satisfaction_Score         92982 non-null  float64
19  Resigned                            92982 non-null  bool
dtypes: bool(1), datetime64[ns](1), float64(2), int64(12), object(4)
memory usage: 14.3+ MB
```

Resultados:

Resumen final

df.describe()

✓ 0.0s

Python

	Employee_ID	Age	Hire_Date	Years_At_Company	Performance_Score	Monthly_Salary	Work_Hours_Per_Week	Projects_Handled	Overtime_Hours	Sic
count	92982.000000	92982.000000	92982	92982.000000	92982.000000	92982.000000	92982.000000	92982.000000	92982.000000	92982
mean	49989.427287	41.014745	2019-09-14 01:21:29.227714560	4.460401	2.997096	6404.305973	44.948227	24.401002	14.535297	7
min	2.000000	22.000000	2014-09-07 08:03:05.556036096	0.000000	1.000000	3850.000000	30.000000	0.000000	0.000000	0
25%	25001.250000	32.000000	2017-03-18 08:03:05.556036096	2.000000	2.000000	5400.000000	38.000000	12.000000	7.000000	3
50%	49998.500000	41.000000	2019-09-19 08:03:05.556036096	4.000000	3.000000	6404.000000	45.000000	24.000000	15.000000	7
75%	74948.750000	50.000000	2022-03-13 08:03:05.556036096	7.000000	4.000000	7200.000000	52.000000	36.000000	22.000000	11
max	100000.000000	60.000000	2024-09-03 08:03:05.556036	10.000000	5.000000	9000.000000	60.000000	49.000000	29.000000	14
std	28857.808013	10.904000	NaN	2.813627	1.373266	1330.992928	8.672218	14.179277	8.491088	4

Tabla de datos faltantes final

porcentaje\_datos\_faltantes\_final = (df.isnull().mean() \* 100).round(2).astype(str) + '%'

print(porcentaje\_datos\_faltantes\_final)

✓ 0.0s

Employee_ID	0.0%
Department	0.0%
Gender	0.0%
Age	0.0%
Job_Title	0.0%
Hire_Date	0.0%
Years_At_Company	0.0%
Education_Level	0.0%
Performance_Score	0.0%
Monthly_Salary	0.0%
Work_Hours_Per_Week	0.0%
Projects_Handled	0.0%
Overtime_Hours	0.0%
Sick_Days	0.0%
Remote_Work_Frequency	0.0%
Team_Size	0.0%
Training_Hours	0.0%
Promotions	0.0%
Employee_Satisfaction_Score	0.0%
Resigned	0.0%
dtype: object	

## Comprobación de que no hay duplicados ni valores inválidos

```
#Verificamos que ya no hay valores invalidos

for c in lista_columnas:
    print(f'En la columna {c} los bbb son: {df[df[c] == 'bbb'].shape[0]}')

En la columna Employee_ID los bbb son: 0
En la columna Department los bbb son: 0
En la columna Gender los bbb son: 0
En la columna Age los bbb son: 0
En la columna Job_Title los bbb son: 0
En la columna Hire_Date los bbb son: 0
En la columna Years_At_Company los bbb son: 0
En la columna Education_Level los bbb son: 0
En la columna Performance_Score los bbb son: 0
En la columna Monthly_Salary los bbb son: 0
En la columna Work_Hours_Per_Week los bbb son: 0
En la columna Projects_Handled los bbb son: 0
En la columna Overtime_Hours los bbb son: 0
En la columna Sick_Days los bbb son: 0
En la columna Remote_Work_Frequency los bbb son: 0
En la columna Team_Size los bbb son: 0
En la columna Training_Hours los bbb son: 0
En la columna Promotions los bbb son: 0
En la columna Employee_Satisfaction_Score los bbb son: 0
En la columna Resigned los bbb son: 0

df.duplicated('Employee_ID').sum() #Comprobamos que no hay duplicados

0
```

# Análisis exploratorio de datos

## Visión general

df

Python

	Department	Gender	Age	Job_Title	Hire_Date	Years_At_Company	Education_Level	Performance_Score	Monthly_Salary	Work_Hours_Per_Week	Projects_Handle
0	Finance	Male	29	Developer	2024-04-18 08:03:05.556036	0	High School	5	7500	34	3
1	Finance	Male	41	Specialist	2015-10-26 08:03:05.556036	8	High School	3	6404	37	2
2	Marketing	Female	48	Analyst	2016-10-22 08:03:05.556036	7	Bachelor	2	4800	52	1
3	Engineering	Female	36	Specialist	2021-07-23 08:03:05.556036	3	Bachelor	2	4800	38	1
4	IT	Male	43	Manager	2016-08-14 08:03:05.556036	8	High School	3	7800	46	3
...	...	...	...	...	...	...	...	...	...	...	...
92978	Engineering	Female	27	Specialist	2019-07-23 08:03:05.556036	4	Bachelor	3	5400	40	2
92979	Operations	Male	35	Analyst	2017-08-29 08:03:05.556036	4	Bachelor	2	4800	45	1
92980	Sales	Male	41	Specialist	2015-02-01 08:03:05.556036	9	High School	4	6300	59	4
92981	Marketing	Male	33	Technician	2016-10-22 08:03:05.556036	7	Bachelor	3	4550	34	3
92982	Operations	Male	39	Specialist	2020-12-24 08:03:05.556036	3	Bachelor	3	5850	34	3

92983 rows × 19 columns

El data frame con el que estuve trabajando tenía 92983 filas y 19 columnas, los datos recogen distinta información de empleados tales como el departamento, género, edad, puesto de trabajo, etcétera. Siendo un total de registros de 1859660 datos.



## Tipos de variables

```
df.dtypes
```

```
Department          object
Gender              object
Age                 int64
Job_Title            object
Hire_Date            datetime64[ns]
Years_At_Company     int64
Education_Level      object
Performance_Score    int64
Monthly_Salary       int64
Work_Hours_Per_Week  int64
Projects_Handled     int64
Overtime_Hours       int64
Sick_Days            int64
Remote_Work_Frequency int64
Team_Size            int64
Training_Hours       int64
Promotions           int64
Employee_Satisfaction_Score float64
Resigned             bool
dtype: object
```

Los tipos de datos en las columnas son de tipo object almacena líneas de texto, int64 que almacena datos numéricos sin decimales, float64 que almacena datos numéricos que tienen números decimales y booleanos que son datos binarios como verdadero o falso.

En este data frame department, gender, job title, education level describen las características de los empleados en formato de texto. Mientras que age y performance score igualmente describen las características, pero en formato numérico.

Resigned es del tipo booleano y almacena si algún empleado se ha ido de la empresa.

Employee satisfaction score es del tipo float y almacena el nivel de satisfacción que tienen los empleados del empleo.

## Resumen estadístico

	Age	Hire_Date	Years_At_Company	Performance_Score	Monthly_Salary	Work_Hours_Per_Week	Projects_Handled	Overtime_Hours	Sick_Days
count	92983.000000	92983	92983.000000	92983.000000	92983.000000	92983.000000	92983.000000	92983.000000	92983.000000
mean	41.014938	2019-09-14 01:08:27.098683904	4.460428	2.997118	6404.333889	44.948066	24.400977	14.535238	7.016272
min	22.000000	2014-09-07 08:03:05.556036	0.000000	1.000000	3850.000000	30.000000	0.000000	0.000000	0.000000
25%	32.000000	2017-03-18 08:03:05.556036096	2.000000	2.000000	5400.000000	38.000000	12.000000	7.000000	3.000000
50%	41.000000	2019-09-19 08:03:05.556036096	4.000000	3.000000	6404.000000	45.000000	24.000000	15.000000	7.000000
75%	50.000000	2022-03-13 08:03:05.556036096	7.000000	4.000000	7200.000000	52.000000	36.000000	22.000000	11.000000
max	60.000000	2024-09-03 08:03:05.556036	10.000000	5.000000	9000.000000	60.000000	49.000000	29.000000	14.000000
std	10.904101	NaN	2.813624	1.373275	1331.012991	8.672310	14.179203	8.491062	4.198860

Remote_Work_Frequency	Team_Size	Training_Hours	Promotions	Employee_Satisfaction_Score
92983.000000	92983.000000	92983.000000	92983.000000	92983.000000
50.079584	10.009841	49.555747	1.000678	2.998564
0.000000	1.000000	0.000000	0.000000	1.000000
25.000000	5.000000	26.000000	0.000000	2.070000
50.000000	10.000000	50.000000	1.000000	2.998583
75.000000	15.000000	73.000000	2.000000	3.930000
100.000000	19.000000	99.000000	2.000000	5.000000
34.282087	5.385067	27.994939	0.799358	1.116361

En el resumen estadístico se pueden ver varios datos, siendo lo más relevante, la edad en donde el promedio está alrededor de 41 años, con un mínimo de 22 y un máximo de 60 años, siendo que el 75% de los datos se encuentra por debajo de 50 años.

Se puede observar que la fecha más antigua de contratación es de 2014 y la más reciente 2024, es decir que la empresa ha estado operando por 10 años aproximadamente.

En cuanto a performance score se puede observar que la escala de puntaje va de un mínimo de 1 a un máximo de 5, siendo que el promedio de rendimiento es de 3 puntos aproximadamente, se puede observar que el 25% de los empleados tiene un rendimiento menor, es decir casi un cuarto de los empleados presenta un mal rendimiento, mientras que solo el 25% de los empleados tiene un rendimiento mayor a 4.

En las horas semanales trabajadas se puede observar algo bastante curioso, el promedio de horas trabajadas en una semana es de 45, asumiendo que sea una semana laboral de lunes a viernes, significa que trabajan alrededor de 9 horas diarias, sin embargo, se puede observar que la jornada máxima registrada es de 60 horas lo que implicaría que posiblemente en ocasiones trabajen sábados y en una jornada de 10 horas, esto sin contar las horas extra.

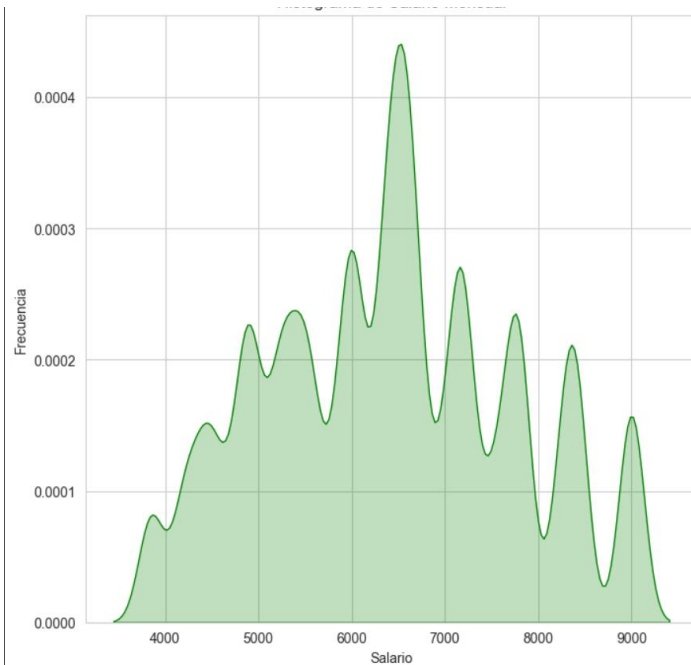
Hablando de horas extra, se puede observar que en promedio trabajan 14 horas extra a la semana, es decir, 2 horas extra diariamente lo que aumentaría la jornada laboral promedio de 9 horas a una de 11 horas en el mejor de los casos ya que se tiene registro de 29 horas extra laboradas en una semana, lo que haría aumentar considerablemente la jornada promedio.

El salario mensual dentro de la empresa se mantiene en un mínimo de 3850 y un máximo de 9000.

Los proyectos manejados en la empresa pueden variar de 0, siendo el mínimo, y un máximo de 50. El 75% de los datos se mantiene por debajo de 36.

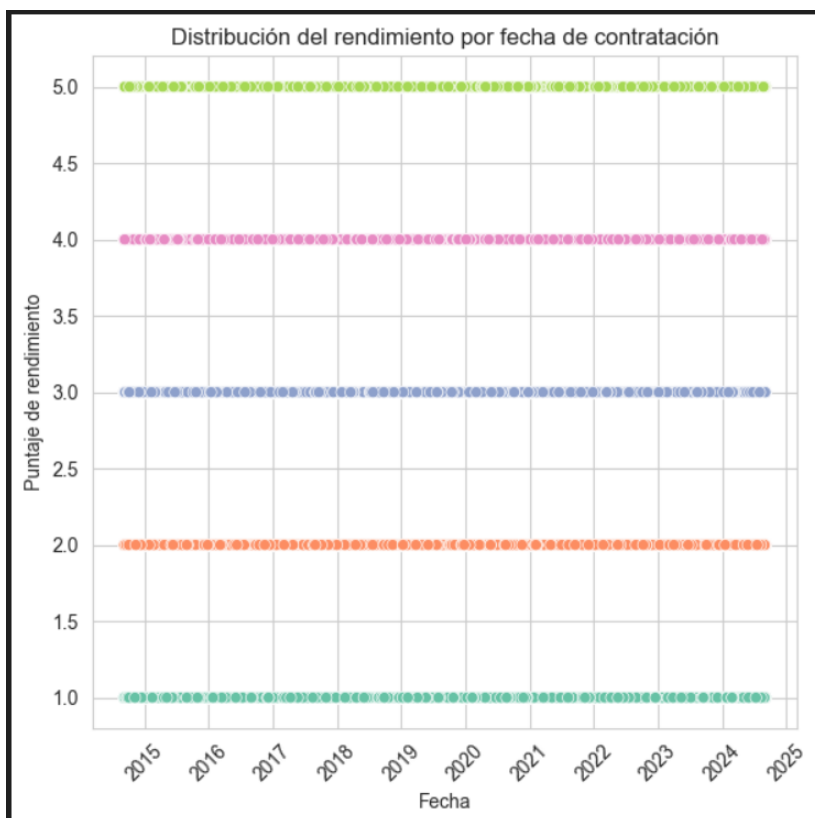
## Visualización y Distribución de Variables Individuales

Gráfico de densidad del salario mensual



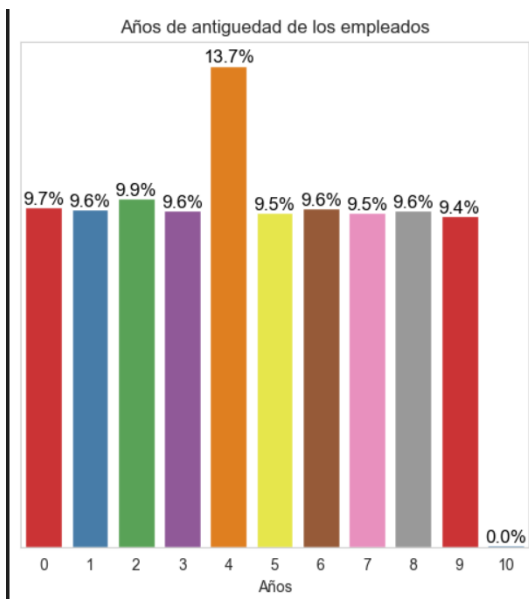
Se puede observar que la mayoría de los salarios mensuales está entre 6000 y 7000, sin embargo, se puede observar que hay una gran cantidad de picos y valles lo que podría indicar que los salarios son muy variables en la empresa.

## Dispersión del rendimiento por año de contratación



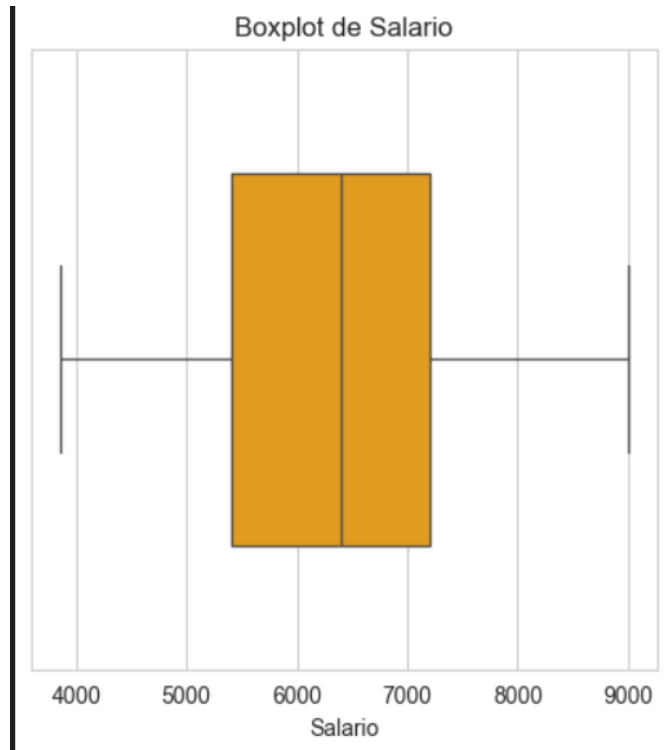
Se puede observar que el rendimiento no se ve afectado por la fecha de contratación.

Gráfico de barras que mide la antigüedad de los empleados



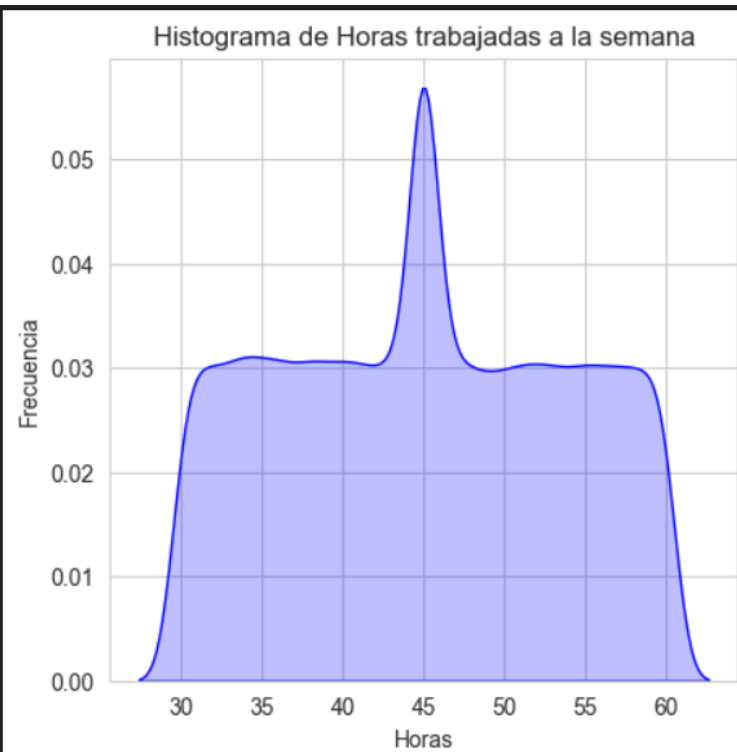
Se puede observar que la mayoría del personal tiene 4 años de antigüedad en la empresa, todos los demás años tienen la misma frecuencia, sin embargo, no hay personal que tenga 10 años en la empresa. Podría indicar que hay un mal ambiente laboral que hace que los empleados renuncien.

Boxplot de salario mensual



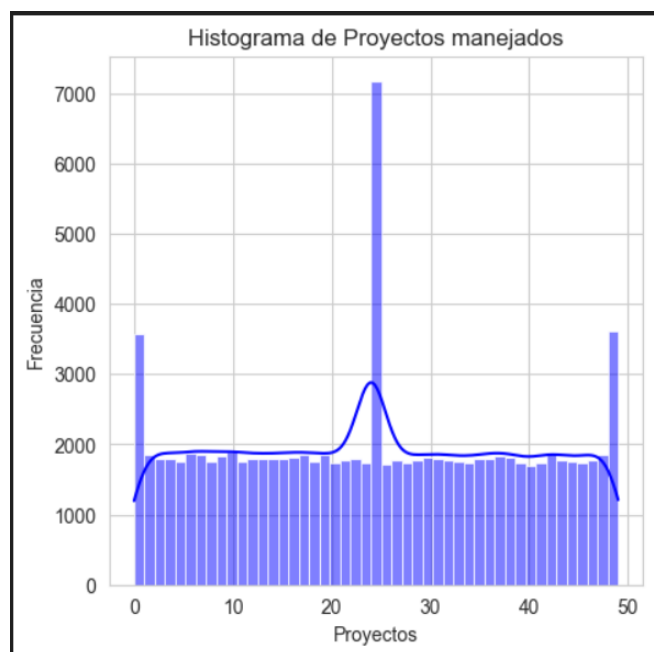
El 50% de los salarios se encuentra entre 6000 y 7000, los bigotes muestran que el salario va de 3900 aproximadamente hasta 9000, no hay datos atípicos.

## Densidad de horas trabajadas semanalmente



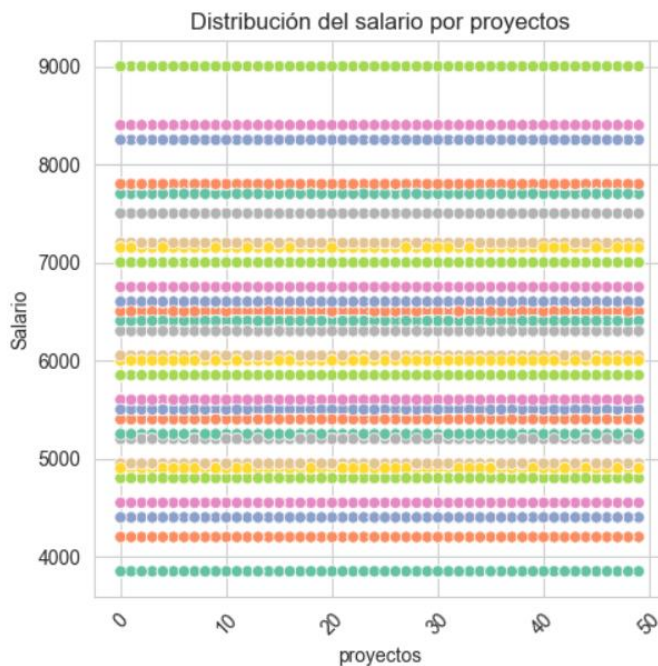
Las horas trabajadas semanalmente están distribuidas uniformemente, sin embargo, se puede observar que hay un pico en 45 horas, lo que indica que hay una porción grande del personal que trabaja más que el resto.

## Histograma de proyectos manejados



Se observa que hay 3 picos en la cantidad de proyectos manejados por empleado, siendo que hay una gran cantidad de personal que maneja entre 20 y 30 proyectos, pero también se puede ver que hay un segmento bastante importante que maneja 50, lo preocupante es que la misma cantidad de empleados maneja 0 proyectos. Esto podría deberse a el puesto de trabajo o a una falta de responsabilidad y compromiso.

#### Distribución de salario por pyoyectos



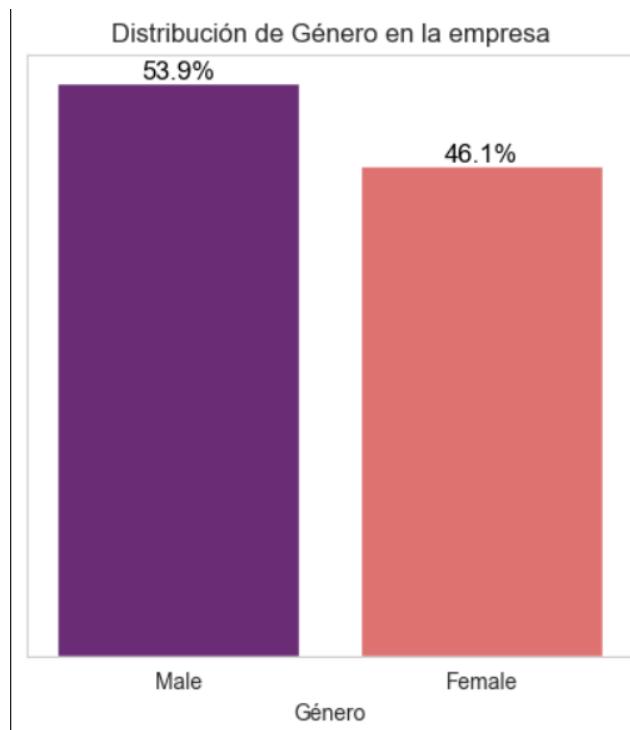
El salario no se ve afectado por el número de proyectos manejados, se puede observar que alguien con 0 proyectos puede ganar desde 3900 hasta 9000 al igual que alguien que ha trabajado en 50 proyectos.

#### Densidad de la satisfacción de los empleados



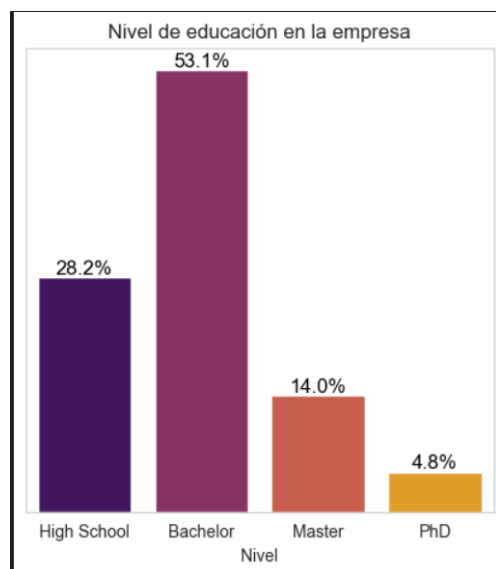
La satisfacción de los empleados tiene una densidad uniforme con un pico en 3. Es decir que la satisfacción no es buena ni mala empresa.

### Género en la empresa



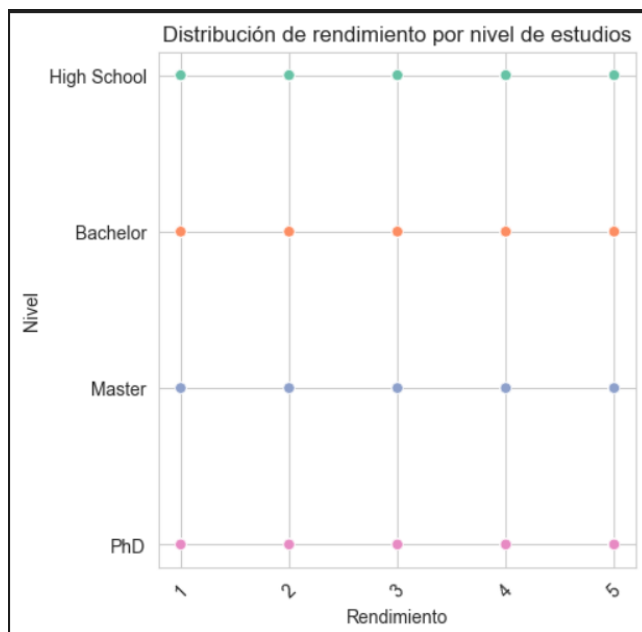
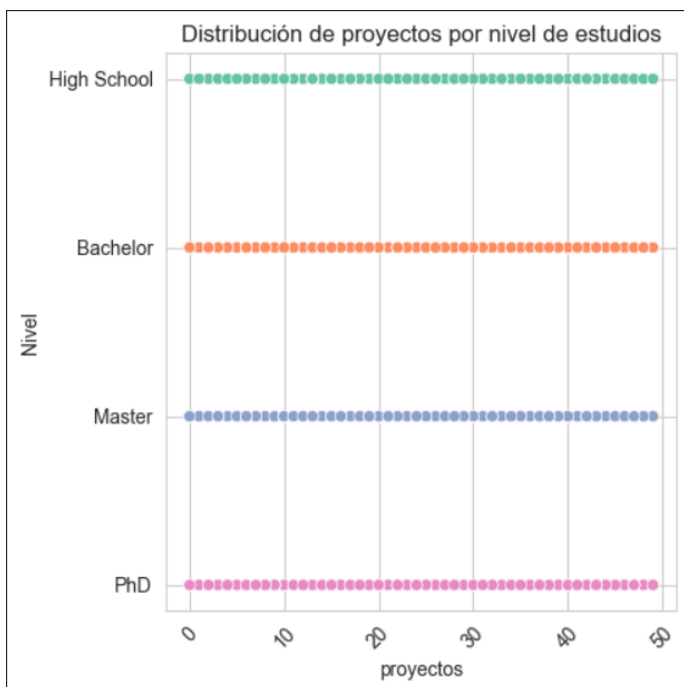
Más de la mitad de la plantilla del personal es masculina.

### Nivel de educación en la empresa

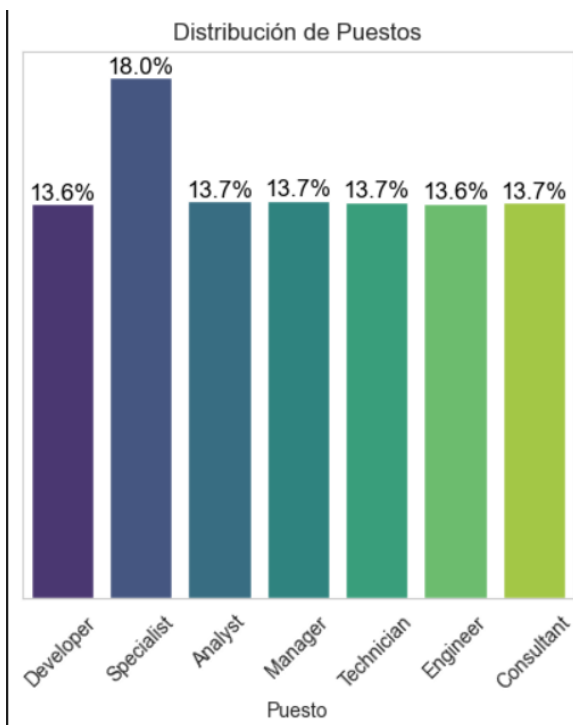




El personal en la empresa en su mayoría tiene el nivel de licenciado y preparatoria, con muy pocos maestros y doctorados. Sin embargo, esto no afecta al rendimiento ni al salario como se muestra en las siguientes gráficas:

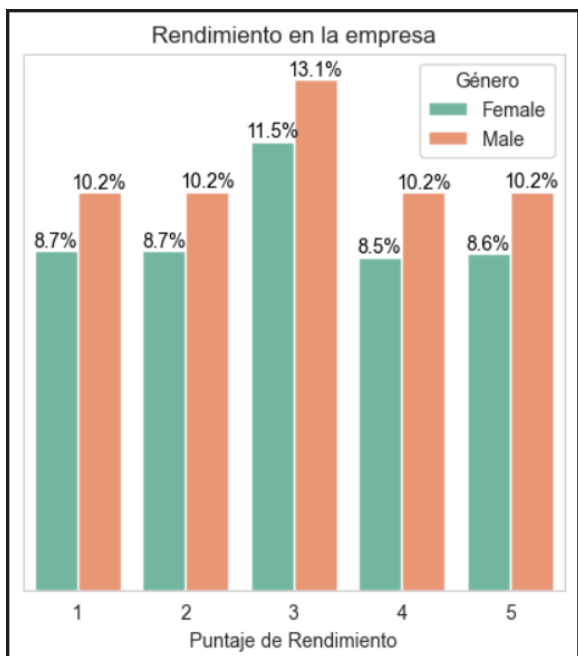


### Distribución de puestos en la empresa



La empresa tiene más empleados como especialistas que en cualquier otro puesto.

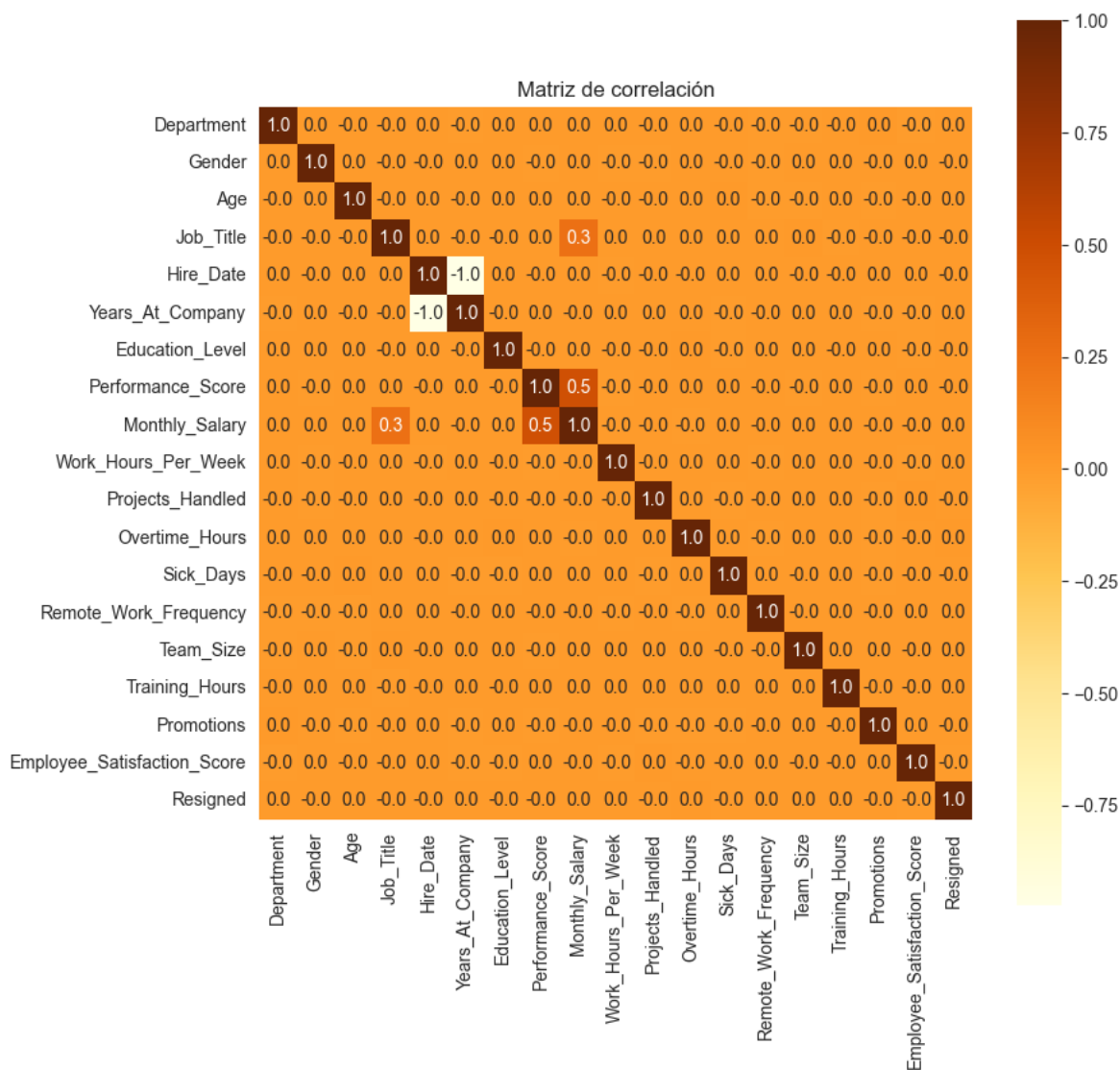
Rendimiento en la empresa por género



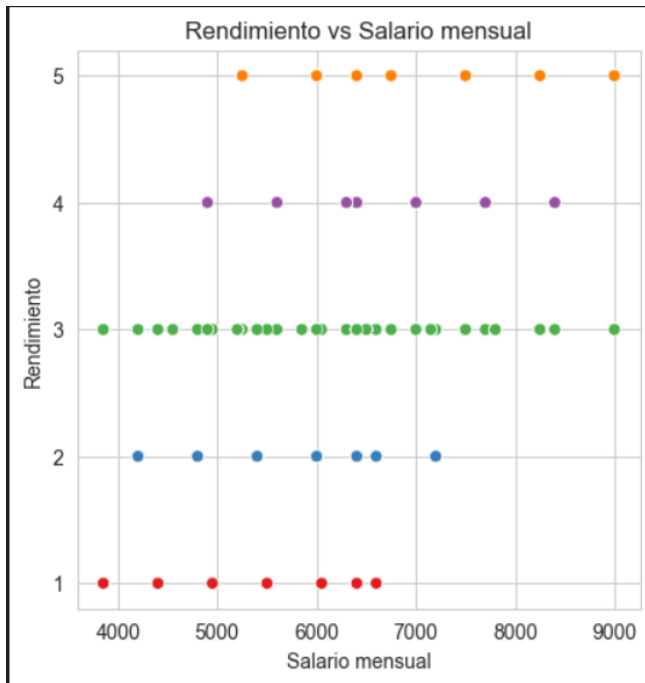
El rendimiento más común en hombres y mujeres es 3 en una escala que va de 1 a 5, se puede observar que la diferencia entre hombres y mujeres es del 2% por categoría, esto solo es una consecuencia de que hay más hombres que mujeres en la empresa.

## Correlación entre Variables

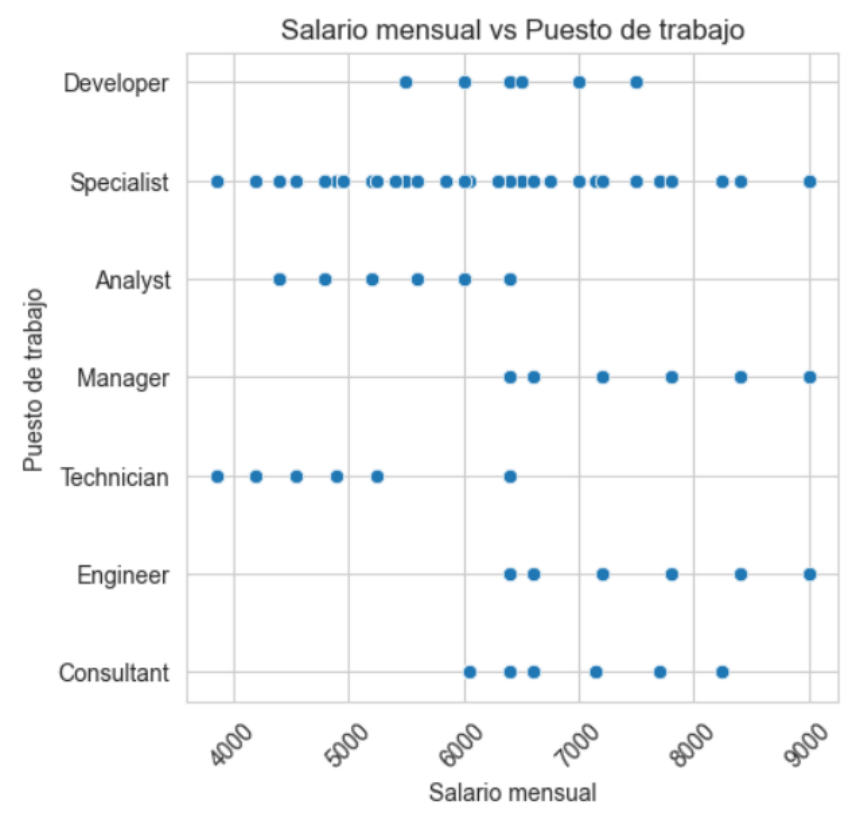
### Matriz de correlación



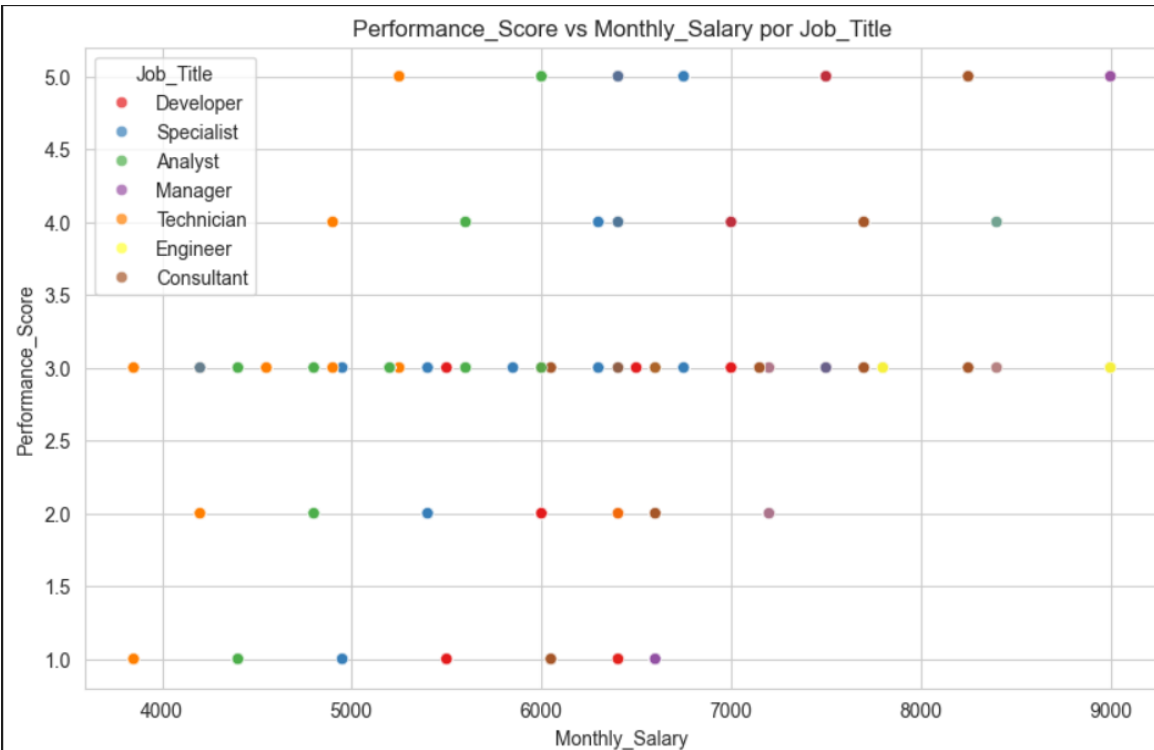
Todos los datos en el dataframe no tienen correlación, a excepción de el puntaje de rendimiento y el salario mensual. Sin embargo, es muy baja lo que significa que no es factible hacer un modelo de regresión lineal por lo que decidí hacer un árbol de decisión.



Se puede observar que distintos sueldos se encuentran agrupados en un rendimiento de 3 puntos. Se puede observar que quienes tienen rendimiento de 1 y 2 se encuentran, en su mayoría, en un rango de 3900 a 6500-6400, esto sugiere que quienes tienen un sueldo menor al promedio suelen tener un rendimiento más bajo. A su vez, se puede observar que quienes tienen un rendimiento mayor a 3 están un poco más agrupados a la derecha donde se encuentran los salarios más altos. Sin embargo, esto no puede ser tomado como un indicador de que el salario determina el rendimiento ya que se puede observar que distintos salarios están en un rendimiento de 3 e incluso que salarios que entran dentro del rango de bajo rendimiento presentan un rendimiento alto.



Se puede observar que el puesto de especialista abarca la mayoría de los salarios, esto se debe a que es el puesto de trabajo que tiene más personal por lo que se puede asumir que es el puesto de trabajo con el rendimiento más variado.



En esta gráfica se puede observar que el puesto de trabajo no necesariamente es un indicador de un buen rendimiento ya que hay puestos que suelen tener un salario mayor al promedio y aun así presentan un bajo rendimiento.

## Análisis de Valores Faltantes

Para la limpieza de valores faltantes impute los valores con la moda de cada columna.

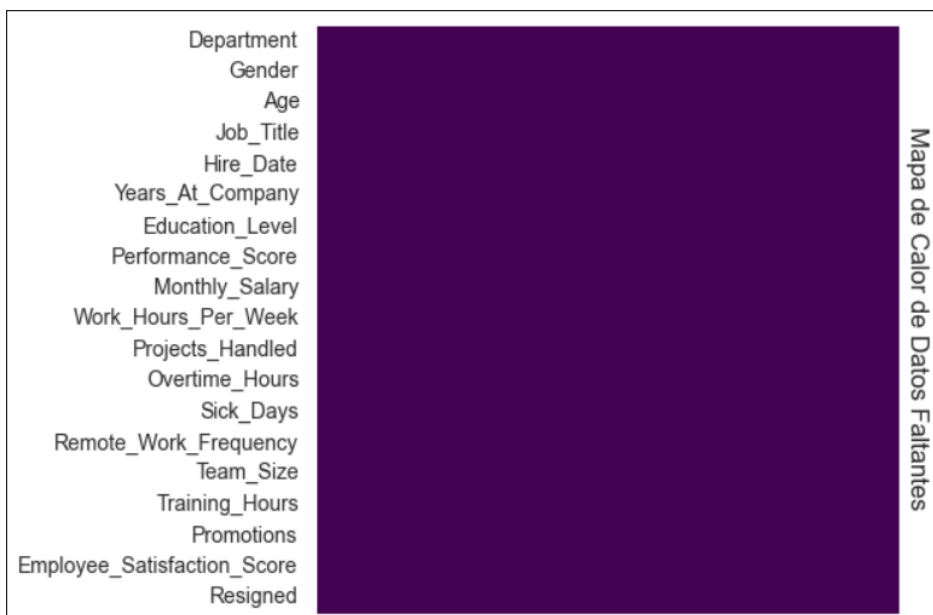
```
categorical_columns = ['Department', 'Job_Title', 'Education_Level', 'Gender']
for col in categorical_columns:
    df[col] = df[col].replace(['Sin dato', 'Other'], pd.NA)

# Imputar los valores faltantes con la moda de cada columna
for col in categorical_columns:
    mode_value = df[col].mode().iloc[0] # Obtener la moda
    df[col].fillna(mode_value, inplace=True)

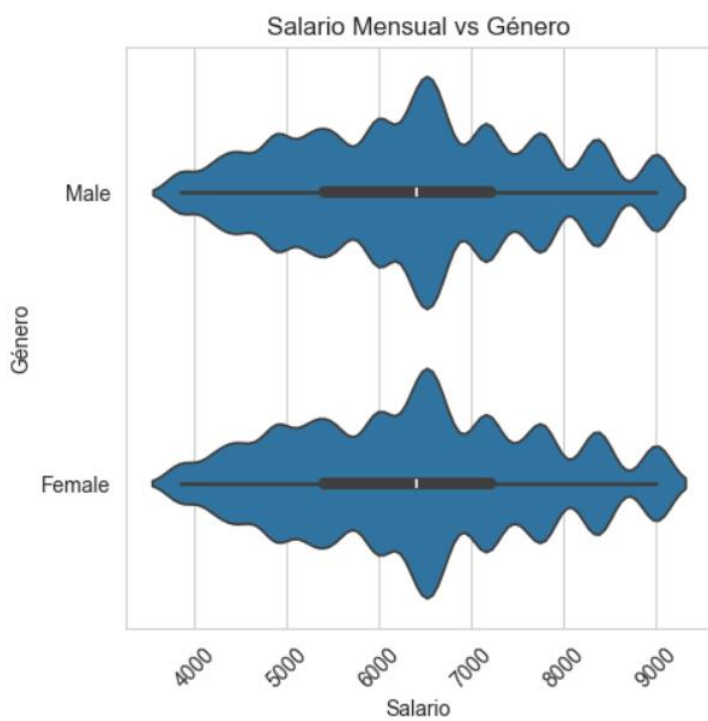
# Confirmar que los valores han sido reemplazados
print(df[categorical_columns].isnull().sum())
```

✓ 0.0s

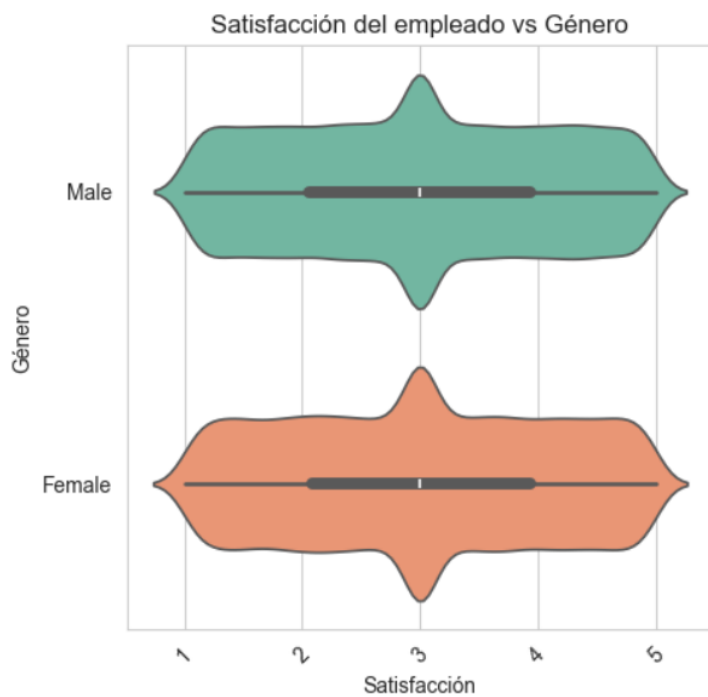
Una vez aplicado el código, este es el mapa de calor de datos faltantes.



## Relación entre Variables Categóricas y Numéricas



Se puede observar que los salarios no difieren entre géneros.



Se puede observar que la satisfacción del empleado tiene la misma distribución tanto en hombres como en mujeres.

## Observaciones y Hallazgos Importantes

El rendimiento está influido, en cierta medida, por el salario y el puesto de trabajo. Según los gráficos de dispersión, quienes tienen un salario más alto suelen presentar un mejor rendimiento. En cuanto al puesto de trabajo, aunque la correlación directa es casi inexistente, podría asumirse una relación indirecta. Esto se debe a que algunos puestos conllevan mayores responsabilidades, pero estas no siempre se reflejan en el salario, lo que podría contribuir a un bajo rendimiento.

Por otro lado, ni las horas semanales trabajadas ni el número de proyectos gestionados parecen impactar el salario de los empleados. Tal como se observó, una persona con 0 proyectos puede ganar lo mismo que otra que maneja 50 proyectos. Este ambiente laboral, puede llegar a ser en extremo cansado y estresante, con jornadas que pueden llegar a ser de más 9 a 11 horas al día sin que esto se vea reflejado en el salario mensual. Sin embargo, como la correlación es muy baja no serán incluidas dentro del modelo.

Para el modelo esto implica que se pueden tomar las características determinantes y así crear un árbol de decisión para poder predecir el rendimiento de futuros empleados con las condiciones que tendrá y poder realizar los ajustes necesarios para lograr un buen rendimiento.

## Modelo de Machine Learning

El modelo que se estará utilizando para este proyecto es un árbol de decisión, el cual se utilizará para predecir el rendimiento de futuros empleados en base a las características identificadas como influyentes. Este modelo nos permitirá optimizar el desempeño laboral.

## Implementación y Entrenamiento

```
data = df_corr[['Performance_Score' , 'Job_Title' , 'Monthly_Salary']]
x = data.drop('Performance_Score' , axis=1)
y = data['Performance_Score']
```

✓ 0.0s

Python

```
#Dividir en entrenamiento y prueba
```

```
x_train , x_test , y_train , y_test = train_test_split(x , y , test_size=0.3 , random_state=42)
```

✓ 0.0s

Python

```
model = DecisionTreeClassifier(random_state=42)
model.fit(x_train , y_train)
```

✓ 0.0s

Python

▼ DecisionTreeClassifier ⓘ ?  
DecisionTreeClassifier(random\_state=42)

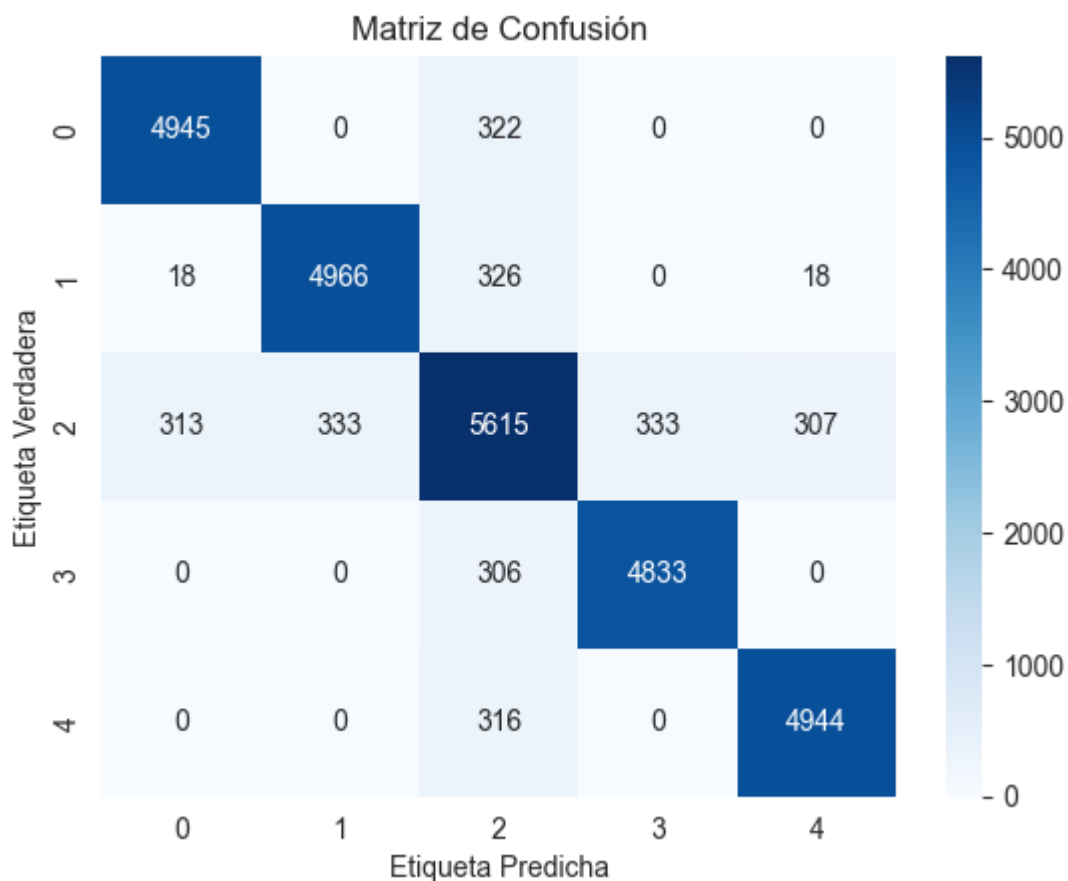
```
y_pred = model.predict(x_test)
```

✓ 0.0s

Python



## Resultados



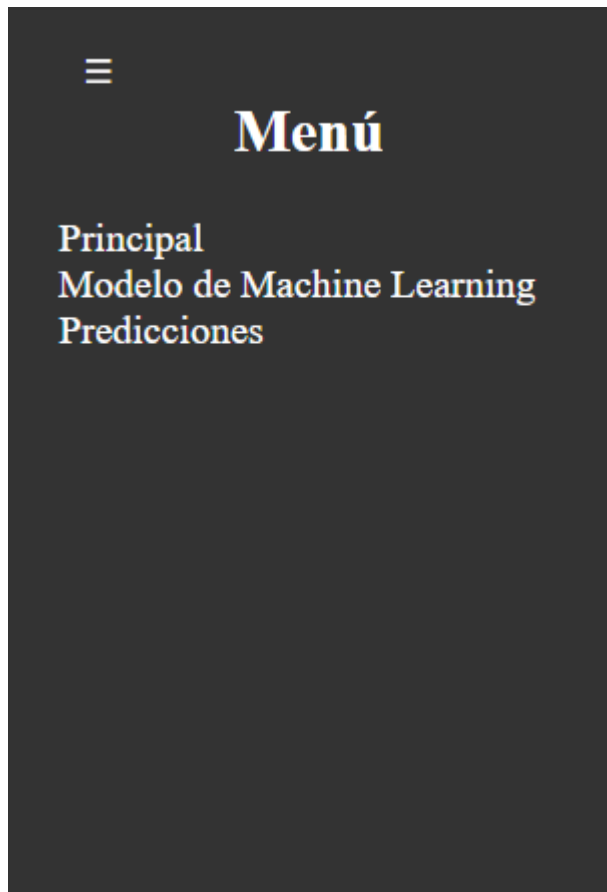
El modelo después del entrenamiento es bastante preciso con muy pocos casos de predicciones erróneas, siendo que la mayoría de los errores se encuentran en la etiqueta 2 que corresponde al nivel de rendimiento 3, esto es provocado por la gran variedad de datos que suele tener ese nivel de rendimiento.

# Dashboard

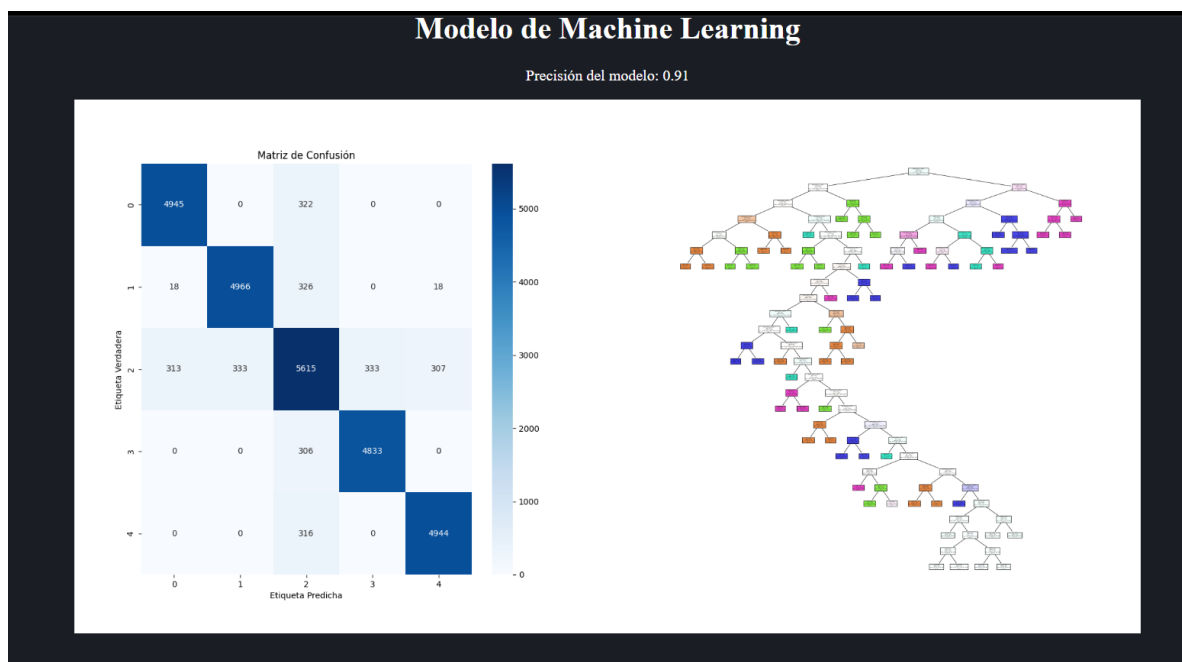


La parte principal del dashboard incluye gráficas relevantes para el análisis tales como las horas trabajadas por semana, el salario mensual vs el rendimiento, los proyectos gestionados y la distribución del salario por proyectos.

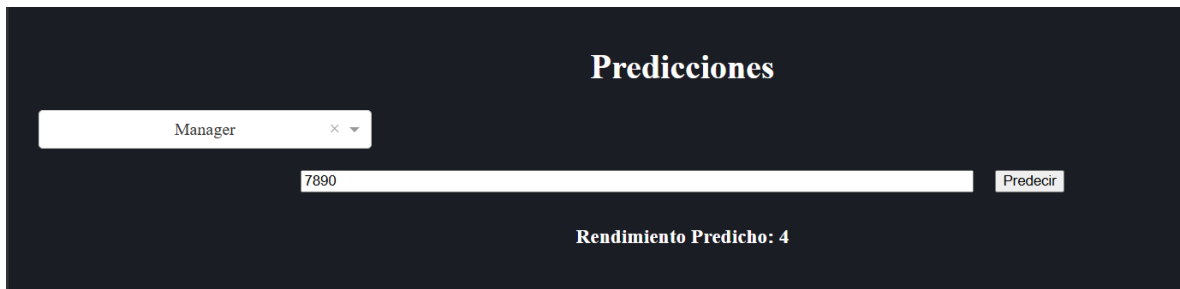
El dashboard también incluye un menú desplegable con las distintas secciones



La sección de Machine Learning incluye información sobre el modelo, tal como la precisión y el árbol graficado



Finalmente, la sección de predicciones tiene una interfaz sencilla de captura de datos.



El dashboard puede ser de utilidad para los clientes y usuarios ya que presenta información importante sobre el rendimiento de los empleados, enfocada desde distintos puntos además de una sección para poder hacer predicciones.

## Conclusiones y Futuras Líneas de Trabajo

El objetivo principal de este proyecto era analizar el rendimiento de los empleados y cuáles eran las características de los empleados que solían tener un rendimiento mayor, logramos encontrar que el salario era un factor que influía en el rendimiento, sin embargo, no es el único factor.

Para poder mejorar el análisis recordaría hacer una recolección de datos más profunda, tales como, qué calificación tienen los empleados de las oficinas, si han sufrido algún tipo de acoso, distancia a la que viven del empleo. Además, recomendaría altamente investigar los sentimientos de los empleados sobre las jornadas laborales, el salario y la distribución de este último.