

PEOPLEMETRICS

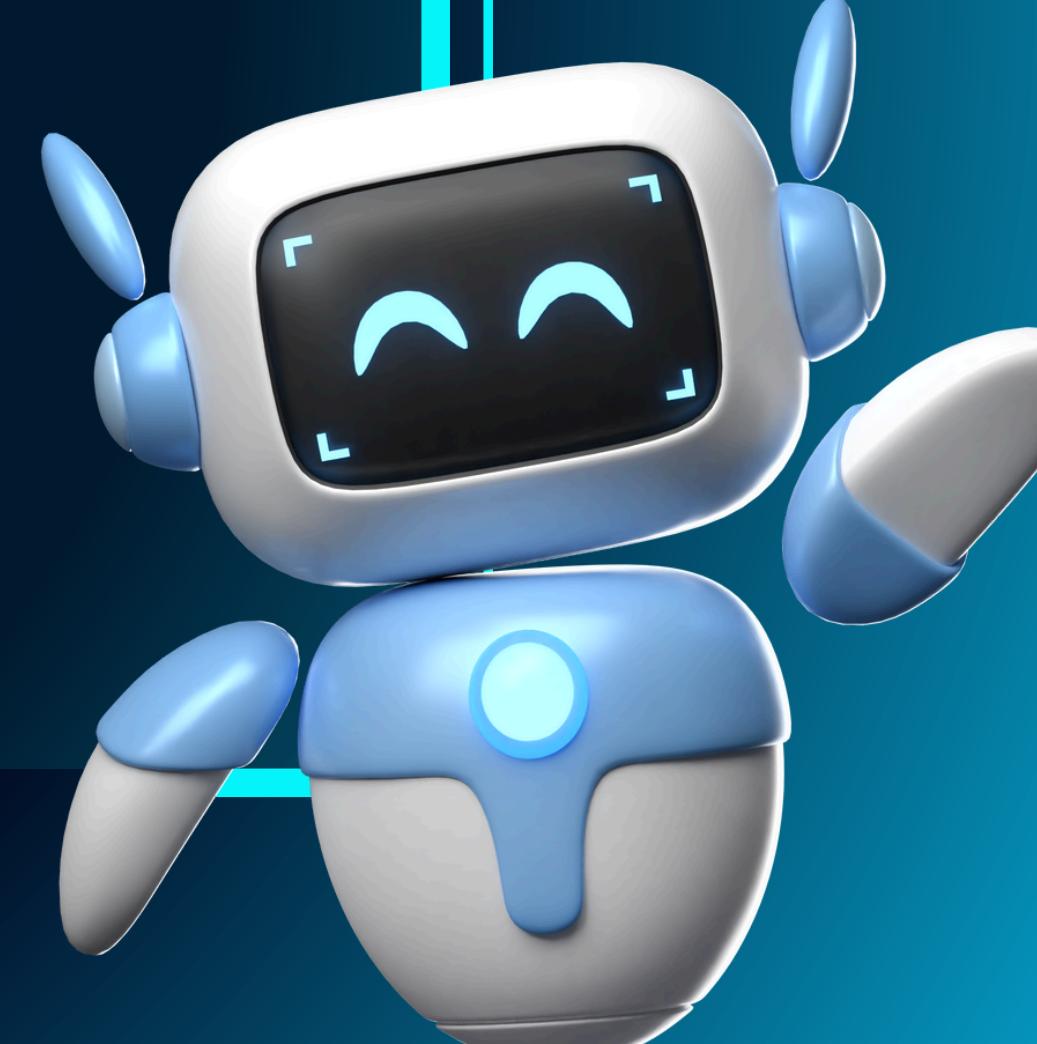
DATAMINDS

OBJETIVOS DEL PROYECTO

CLIENTE : ABC CORPORATION

**Identificar los factores que influyen en la
satisfacción y permanencia de los empleados.**

A través de un análisis exhaustivo de datos proporcionaremos a ABC Corporation información estratégica valiosa para mejorar su toma de decisiones.



EQUIPO DATAMINDS



Product Owner:
LIDIA FRAILE



Scrum Master:
SILVIA MARRERO



Developer:
ISABEL MARTINEZ



Developer:
MARGARITA TORRES



Developer:
SHARON ARMENI





METODOLOGÍA Y PROCESO

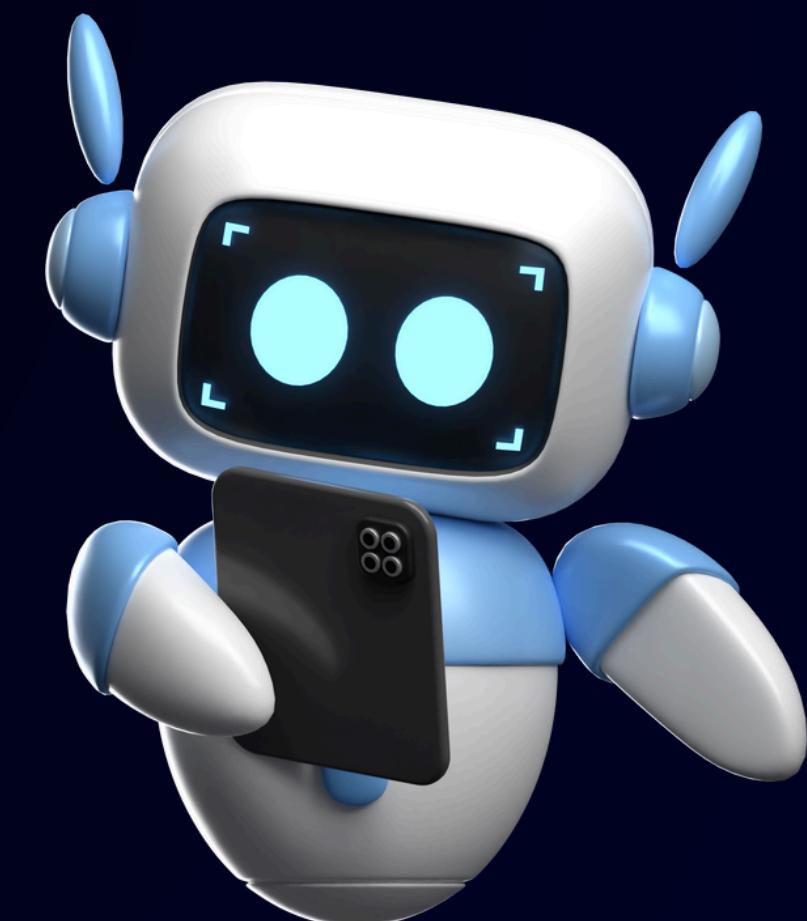
SISTEMA DE TRABAJO

Sistema *Agile* dentro de un marco de trabajo *Scrum*

ORGANIZACIÓN

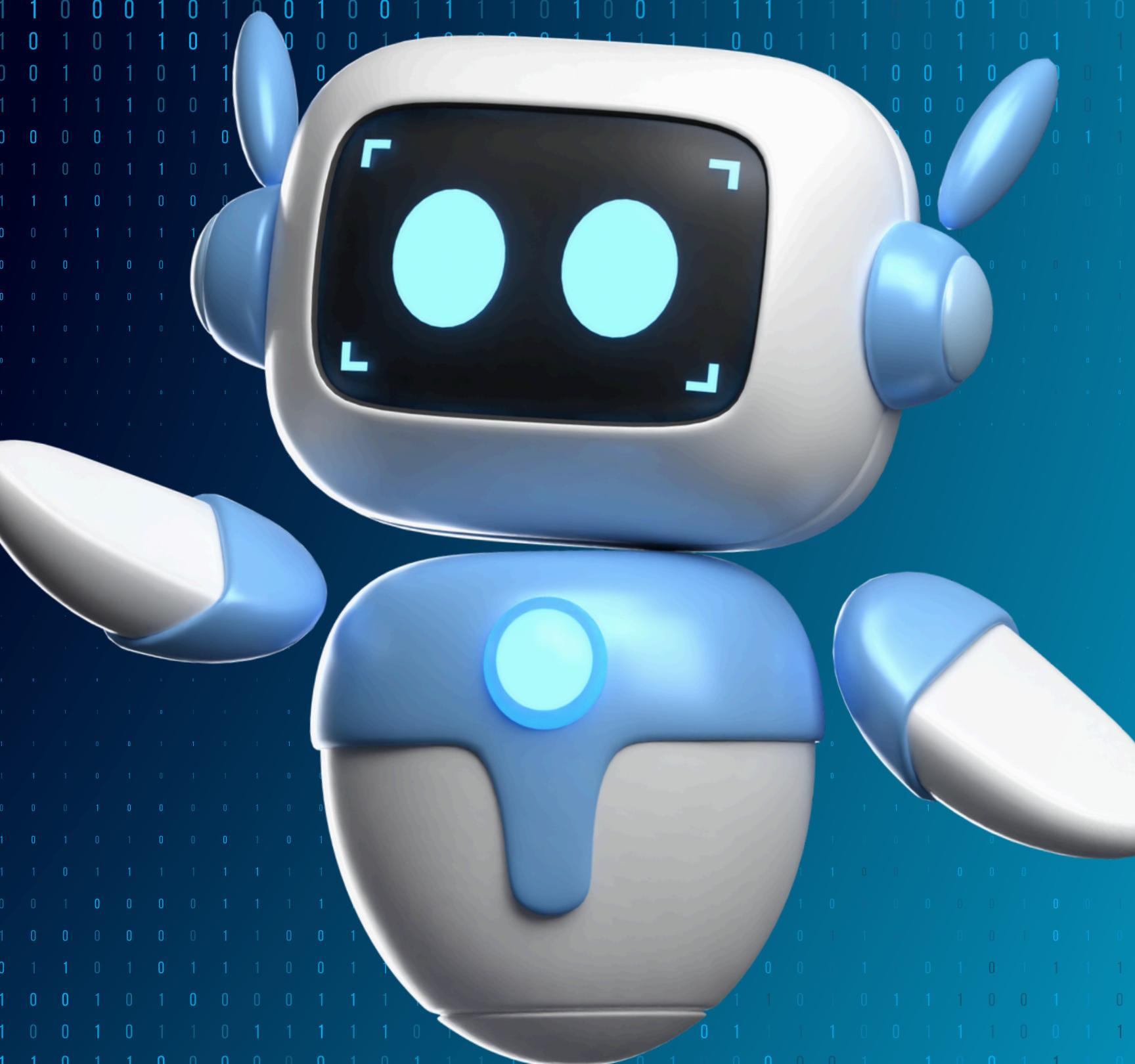
- ✓ 3 semanas
- ✓ 2 sprints
- ✓ 6 fases

HERRAMIENTAS



FASES

PEOPLEMETRICS



ANÁLISIS EXPLORATORIO DE DATOS (EDA)



File Edit Selection View Go Run ... 🔍 proyecto-da-promo-H-modulo-3-team-3-DataMinds [WSL: Ubuntu-20.04] ⚡ X

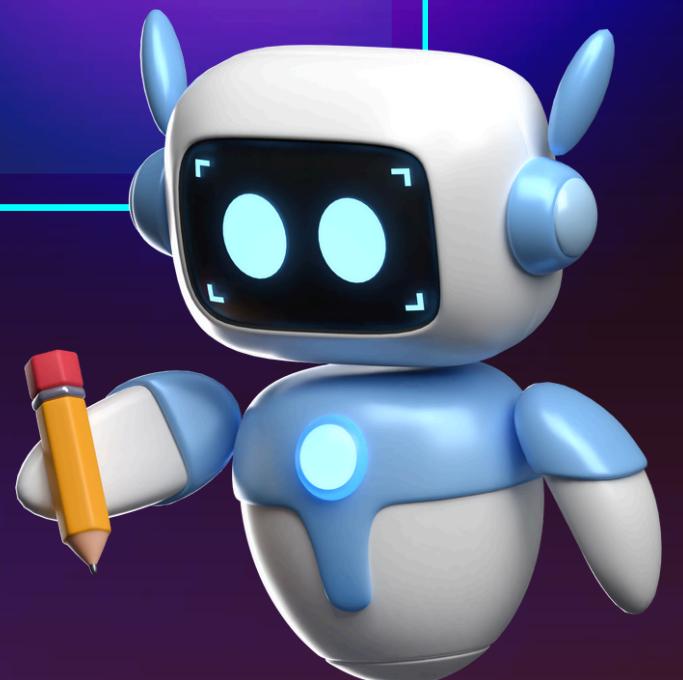
EXPLORER Fase1.py

PROYECTO-DA-PROMO-H... Fase1.py > ... hr_data.csv

```
15 def generate_report(nombre_fichero):  
16     """  
17         if ".csv" in nombre_fichero:  
18             try:  
19                 df=pd.read_csv(nombre_fichero) #Lee el archivo en un DataFrame de pandas  
20             except:  
21                 print("Error while reading the file")  
22                 exit() #SystemExit: si el archivo no se puede leer  
23             elif ".xlsx" in nombre_fichero:  
24                 df=pd.read_excel(nombre_fichero)  
25             else:  
26                 print("No valid extension")  
27                 exit() #SystemExit: si la extensión del archivo no es '.csv' o '.xlsx'  
28  
29  
30             head=df.head() #Mostramos las primeras filas del DataFrame  
31             tail=df.tail() #Mostramos las últimas filas del DataFrame  
32  
33             #Visualizamos la información básica sobre el DataFrame  
34             info=df.info(buf=buffer) # genera info sobre el DataFrame y la almacena en el buffer  
35             info_str=buffer.getvalue() # se extraen los datos del buffer como una cadena para después escribirlos en un archi  
36  
37             #Visualizamos la forma del DataFrame  
38             shape_row=df.shape[0] # número de filas  
39             shape_columns=df.shape[1] # número de columnas  
40  
41             columns_name=df.columns #Mostramos los nombres de las columnas  
42  
43             #Visualizamos en qué columnas hay nulos y cuántos hay
```

WSL: Ubuntu-20.04 ⚡ 0 △ 0 ⌂ 0 Ln 87, Col 32 Spaces: 4 UTF-8 LF { Python ⚡ Select Interpreter

TRANSFORMACIÓN DE LOS DATOS



DISEÑO DE BBDD E INSERCIÓN DE LOS DATOS



The screenshot shows a code editor interface with a dark theme. The title bar reads "proyecto_peoplemetrics [WSL: Ubuntu-20.04]". The left sidebar has icons for File, Edit, Selection, View, Go, Run, etc., and displays the project structure under "EXPLORER". The main area shows a Python script named "bbdd_dataminds_soporte.py".

```
def create_bbdd(query, contraseña):
    mycursor.execute(query)
    print("DataBase successfully created")

except mysql.connector.Error as err:
    print(err)
    print("Error Code:", err(errno))
    print("SQLSTATE", err.sqlstate)
    print("Message", err.msg)

# %%
def create_table(usuario, contraseña, bbdd, query):
    """
    Esta funcion permite crear una tabla en una BBDD en MySQL

    Args:
    - usuario: usuario para la conexion al servidor
    - contraseña: contraseña para la conexión al servidor
    - bbdd: nombre de la bbdd donde queremos crear la tabla
    - query: Consulta para la creacion de la tabla
    """

    cnx = mysql.connector.connect(user=usuario, password=contraseña,
                                  host='127.0.0.1', database=bbdd)

    mycursor = cnx.cursor()

    try:
        mycursor.execute(query)
        print("Table created")
```

The status bar at the bottom shows "Ln 1, Col 1" and "Spaces: 4". The bottom right corner has a "Select Interpreter" button.

CREACIÓN DE UNA ETL



AUTOMATIZACIÓN

**PERMITE EL ANÁLISIS DE
TENDENCIAS A LO LARGO DEL
TIEMPO**

**LIMPIEZA Y
PREPARACIÓN**

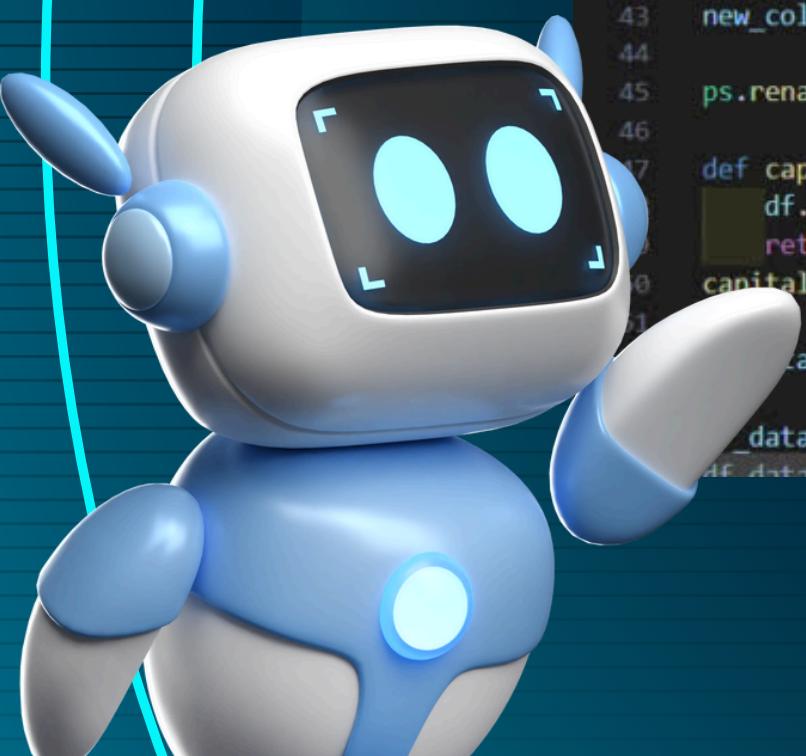
**FACILITA LA INTEGRACIÓN DE LOS
DATOS EN SISTEMAS Y FUENTES**

**MANTENIMIENTO DE LA
CALIDAD**

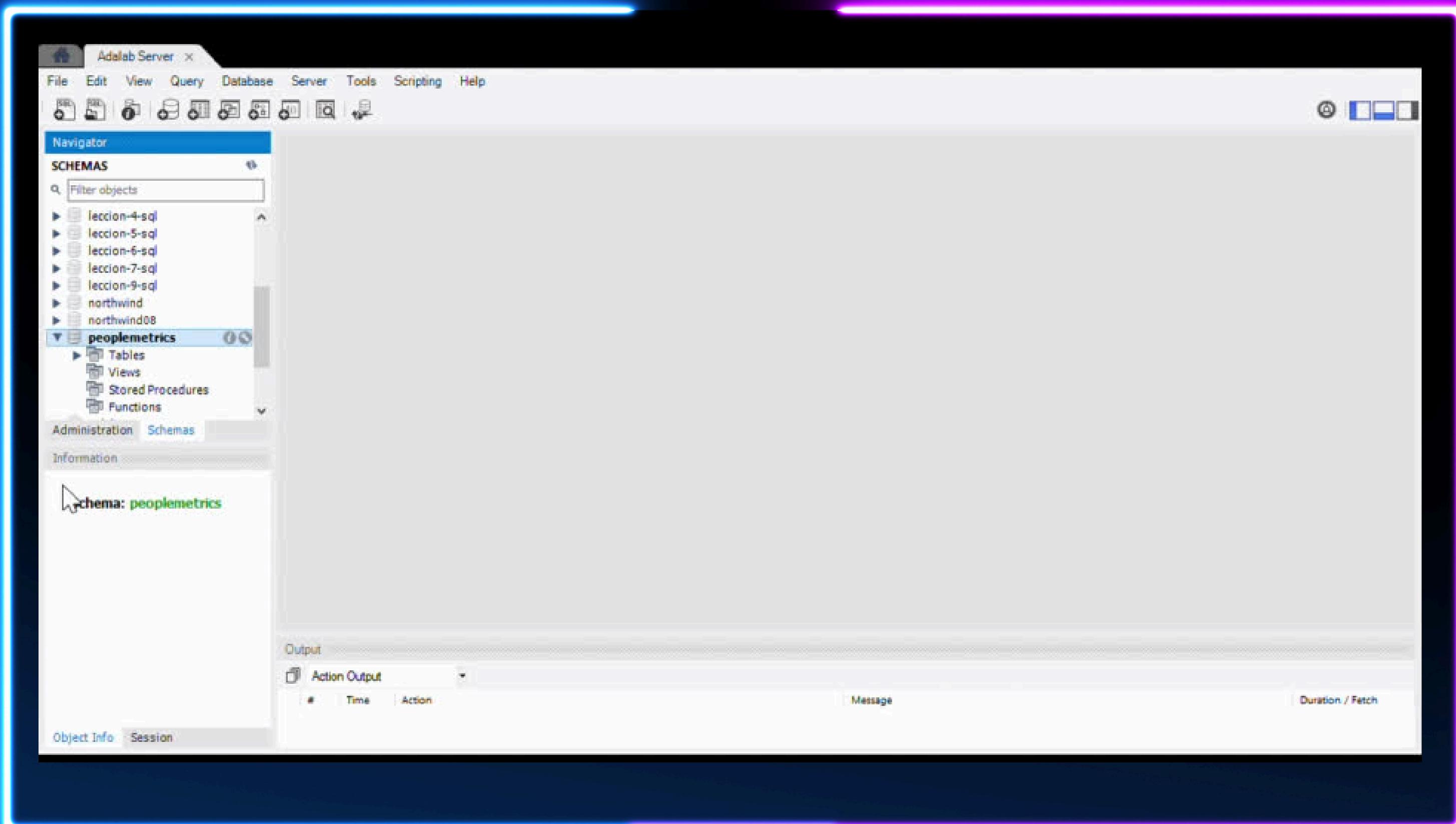
**GARANTIZA QUE LOS DATOS
ESTÉN DISPONIBLES**

**AHORRA TIEMPO Y
RECURSOS**

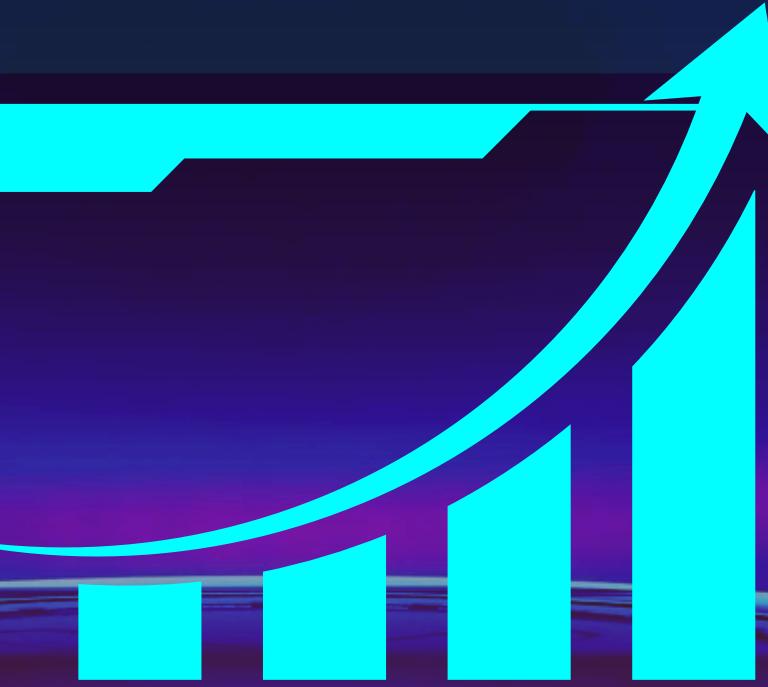


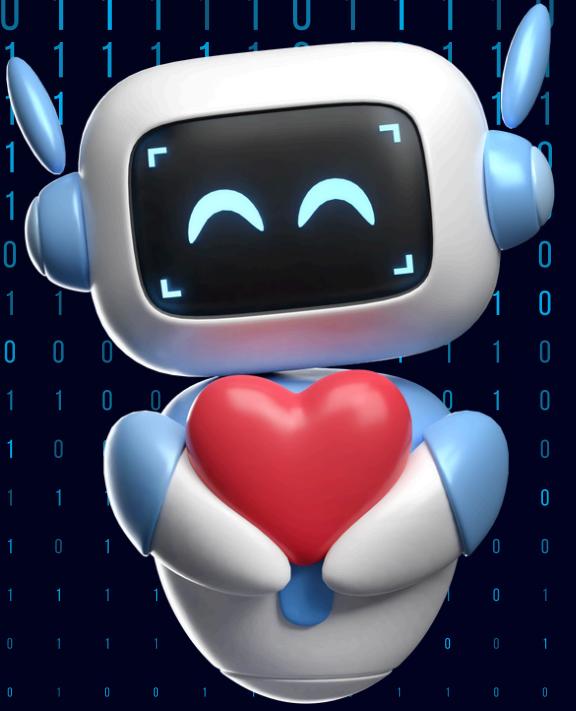


```
main_proyecto.py • proyecto_soporte.py 4
proyecto-da-promo-H-modulo-3-team-3-DataMinds > fase_5 > main_proyecto.py > ...
19 print('-----\n')
20 print('          Initial data exploration\n')
21 print('-----\n')
22 #Sacamos un informe inicial de los datos brutos que tenemos.
23 url1='https://raw.githubusercontent.com/s-armeni/proyecto-da-promo-H-modulo-3-team-3-DataMinds/main/HR%20RAW%20DATA.csv'
24 ps.initial_exploration(url1)
25
26
27
28 print('-----\n')
29 print('          Exploratory Data Analysis\n')
30 print('-----\n')
31
32
33 #Importamos csv.
34 url1='https://raw.githubusercontent.com/s-armeni/proyecto-da-promo-H-modulo-3-team-3-DataMinds/main/HR%20RAW%20DATA.csv'#utilizamos url para que sea reproducible en otra maquina
35
36 df_data=ps.import_csv(url1)
37 #Visualizamos el número total de filas y de columnas
38 print(f'Cantidad de filas: {df_data.shape[0]}\nCantidad de columnas: {df_data.shape[1]}' )
39
40 #limpiamos las columnas.
41 #Al haberlas arreglado parcialmente a mano, importamos la lista con las nuevas columnas.
42 data = open("columns_handmade.txt", "r")#Esta funcion no es parte del soporte por lo que no hace falta que lleve api
43 new_columns = list(data)
44
45 ps.rename_columns(df_data,new_columns)
46
47 def capitalize(df):
48     df.columns=df.columns.str.capitalize()
49     return df
50 capitalize(df_data)
51
52 df_data['Age']=df_data['Age'].apply(ps.age_to_int)#Estandarización columna 'Age'
53
54 df_data.loc[df_data['Distance from home']<0, 'Distance from home']*= -1#Eliminamos los '-' de 'Distance_from_home'
55 df_data['Marital status'].replace(['Married', 'Unmarried'])#Corregimos errores ortográficos en la columna 'Marital_status'
```



PRUEBA DE A/B TESTING

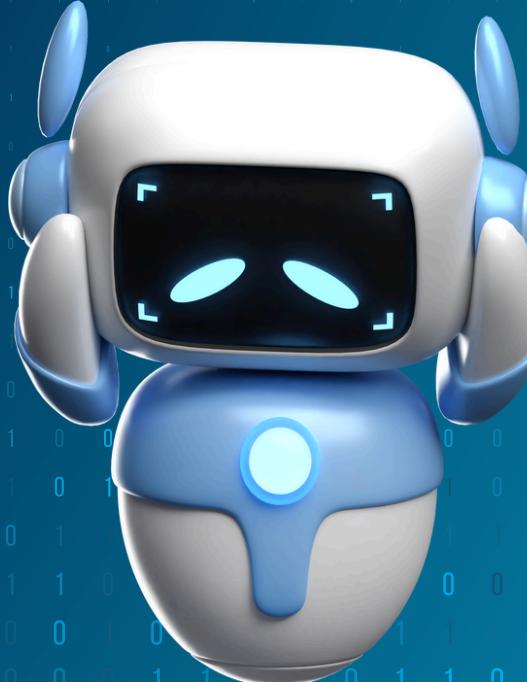


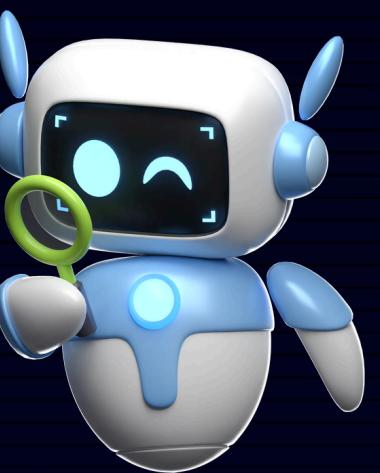


GRUPO A:
EMPLEADOS SATISFECHOS

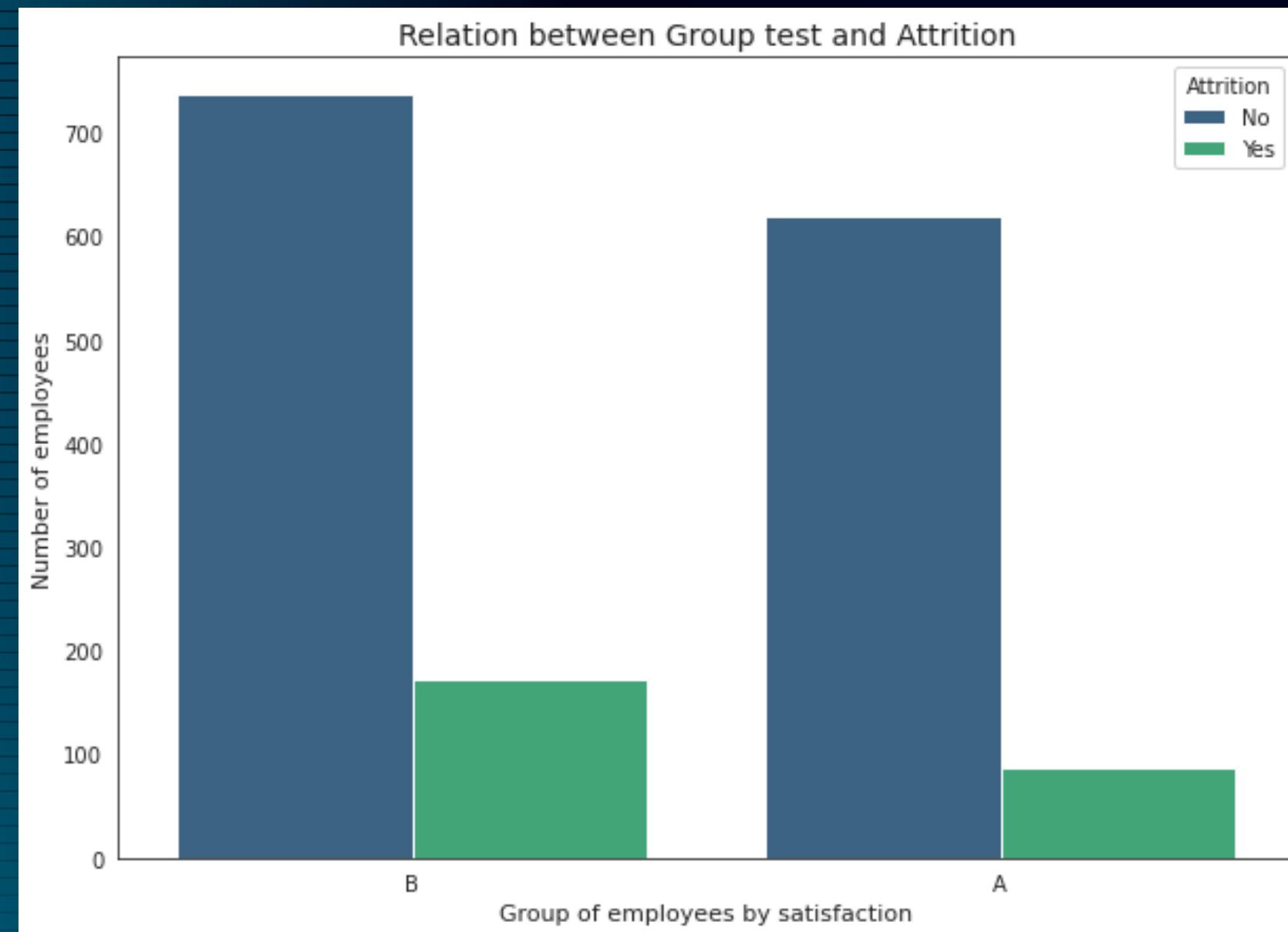
TASA DE ROTACIÓN

GRUPO B:
EMPLEADOS MENOS SATISFECHOS

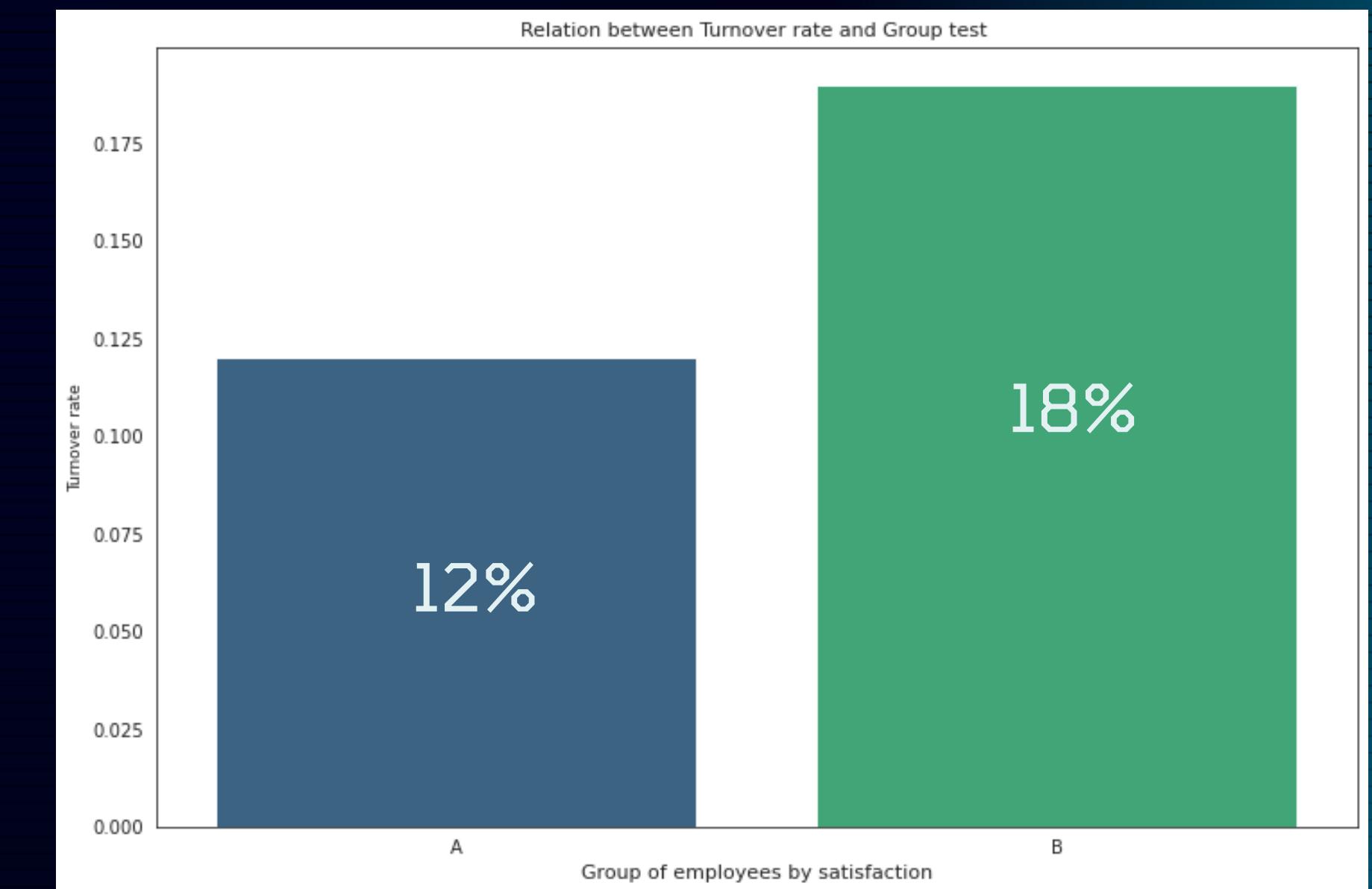




RELACIÓN ENTRE LOS GRUPOS A/B Y LAS PERSONAS QUE HAN DEJADO LA EMPRESA



RELACIÓN ENTRE LOS GRUPOS A/B Y LAS TASA DE ROTACIÓN



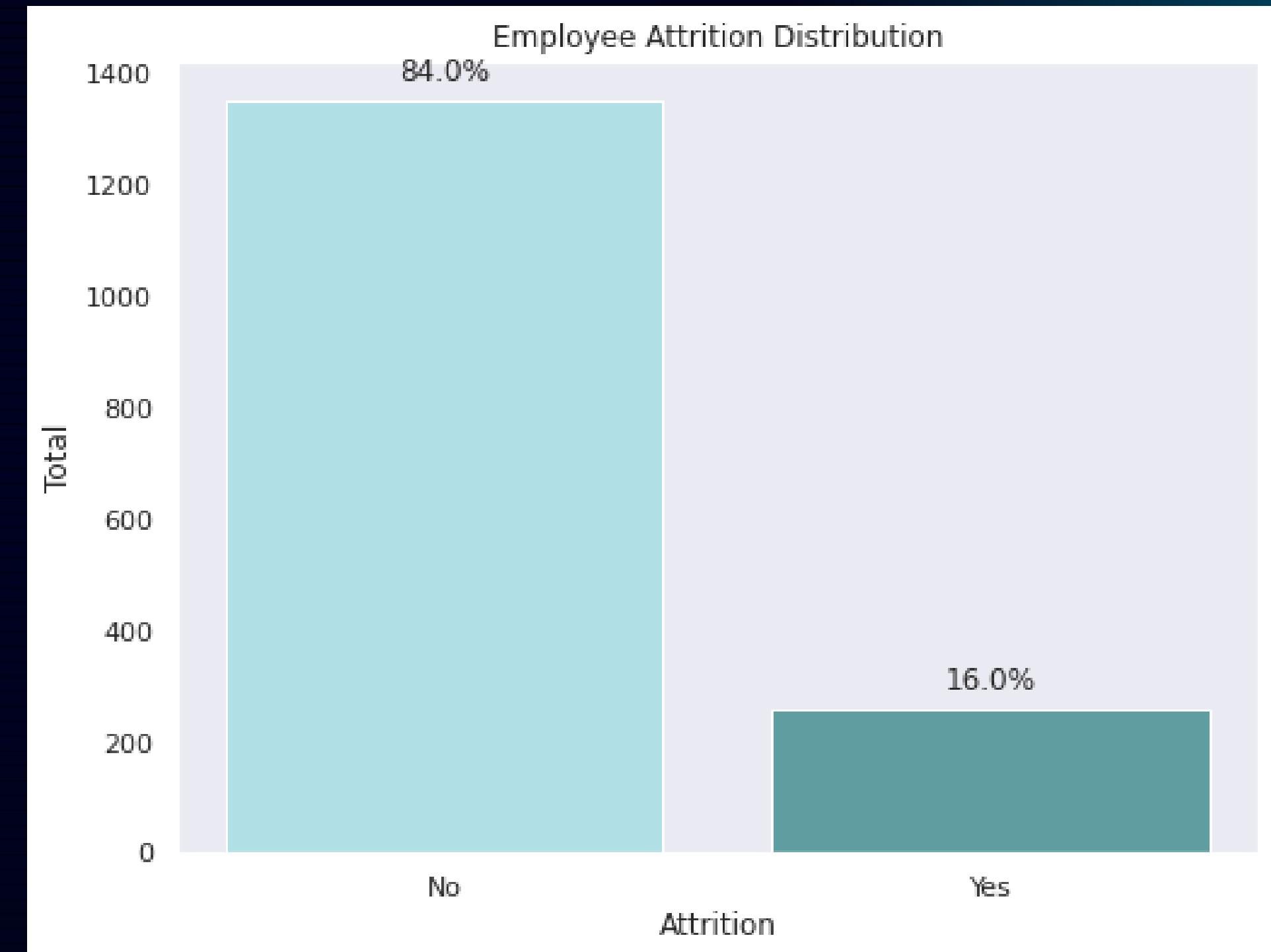
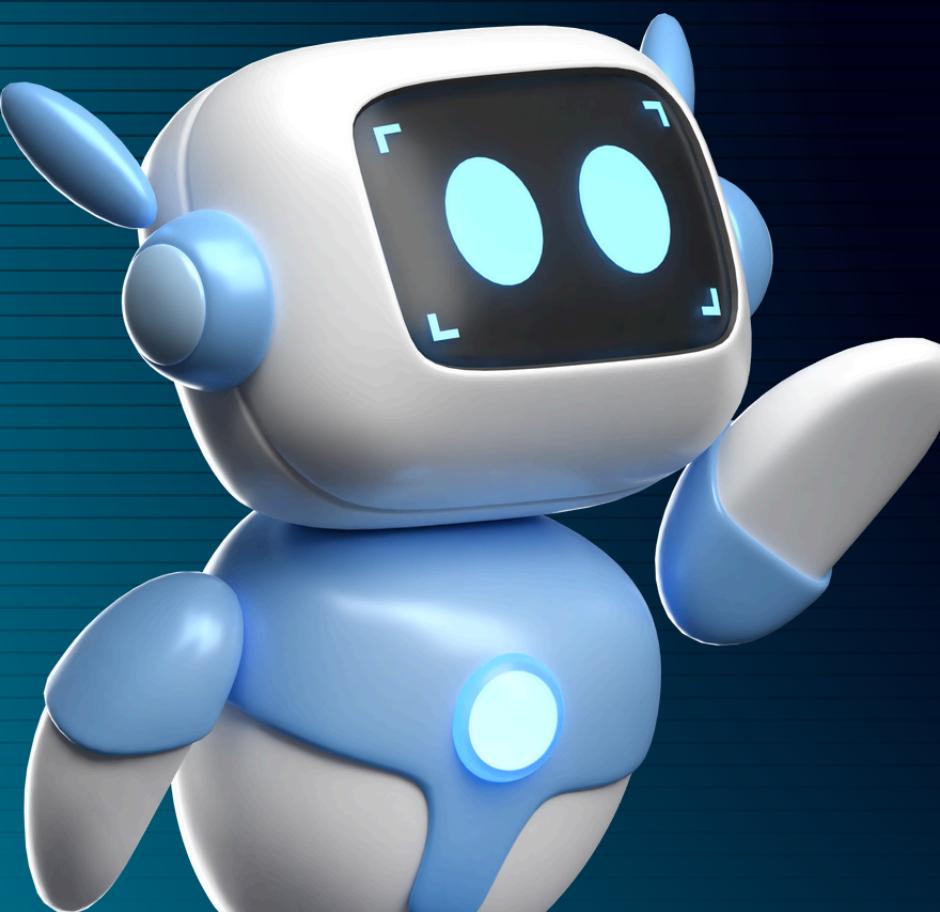
REPORTE DE RESULTADOS



Proporción de bajas

Observaciones

- Baja tasa de abandono.
- Alta retención de empleados.



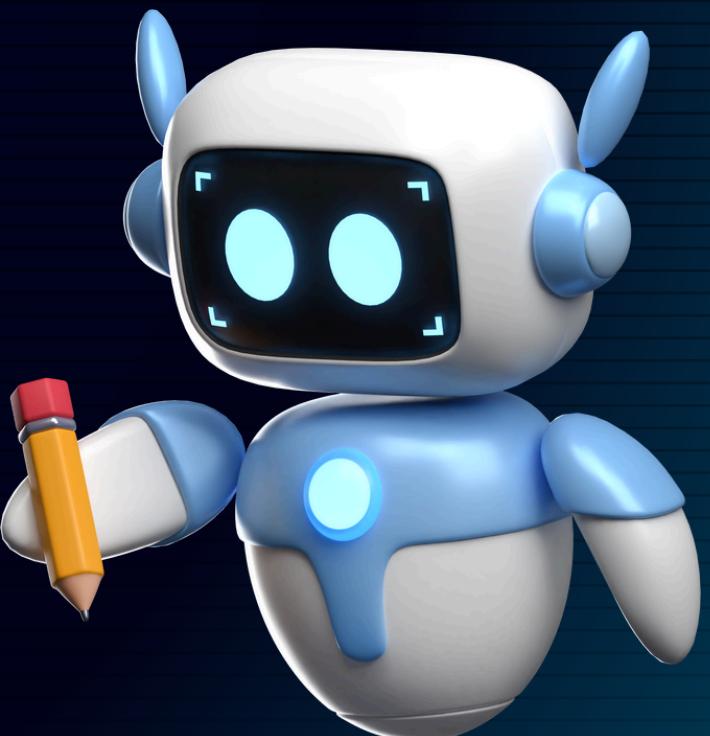
Análisis de la relación entre características individuales y la tase de rotación

Relaciones más relevantes

- Edad.
- Nivel educativo.
- Categoría laboral.
- Aumento salarial.
- Valoración del desempeño.
- Acciones formativas recibidas.
- Conciliación laboral.
- Antigüedad.
- Satisfacción general del trabajador.

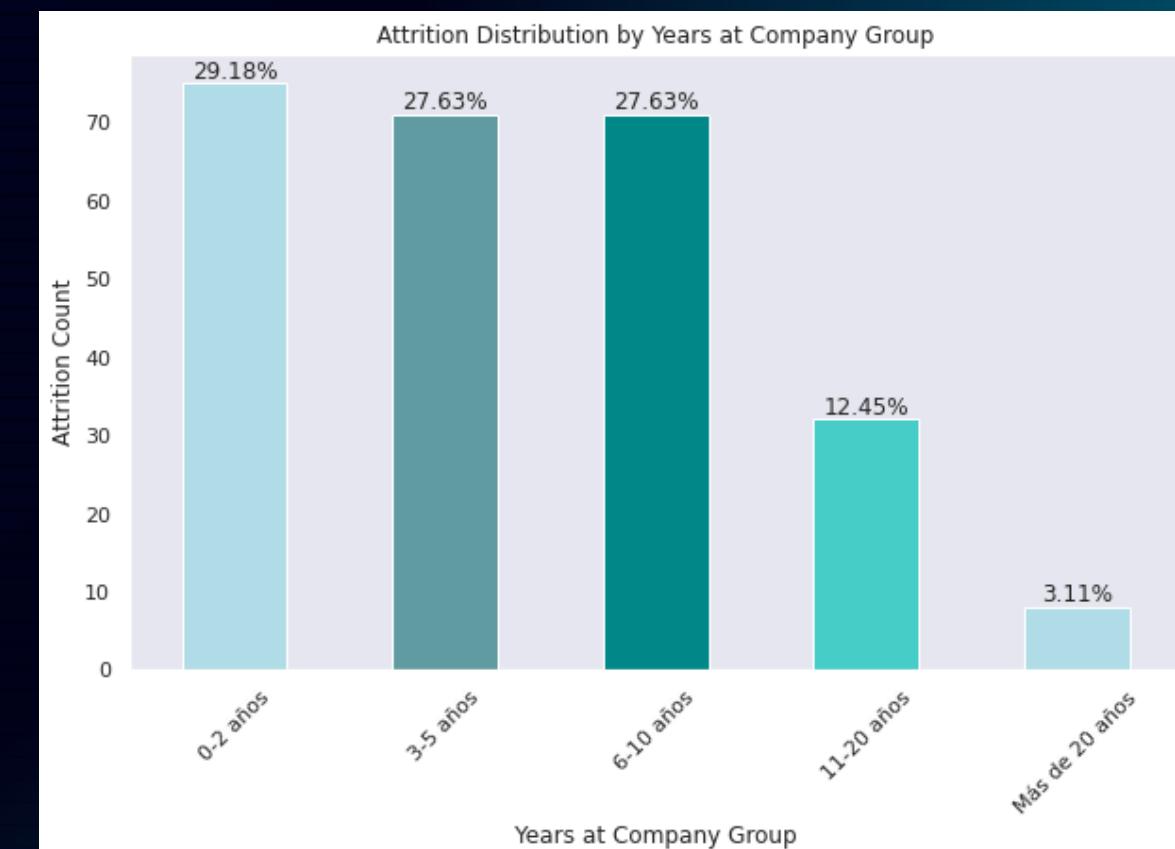
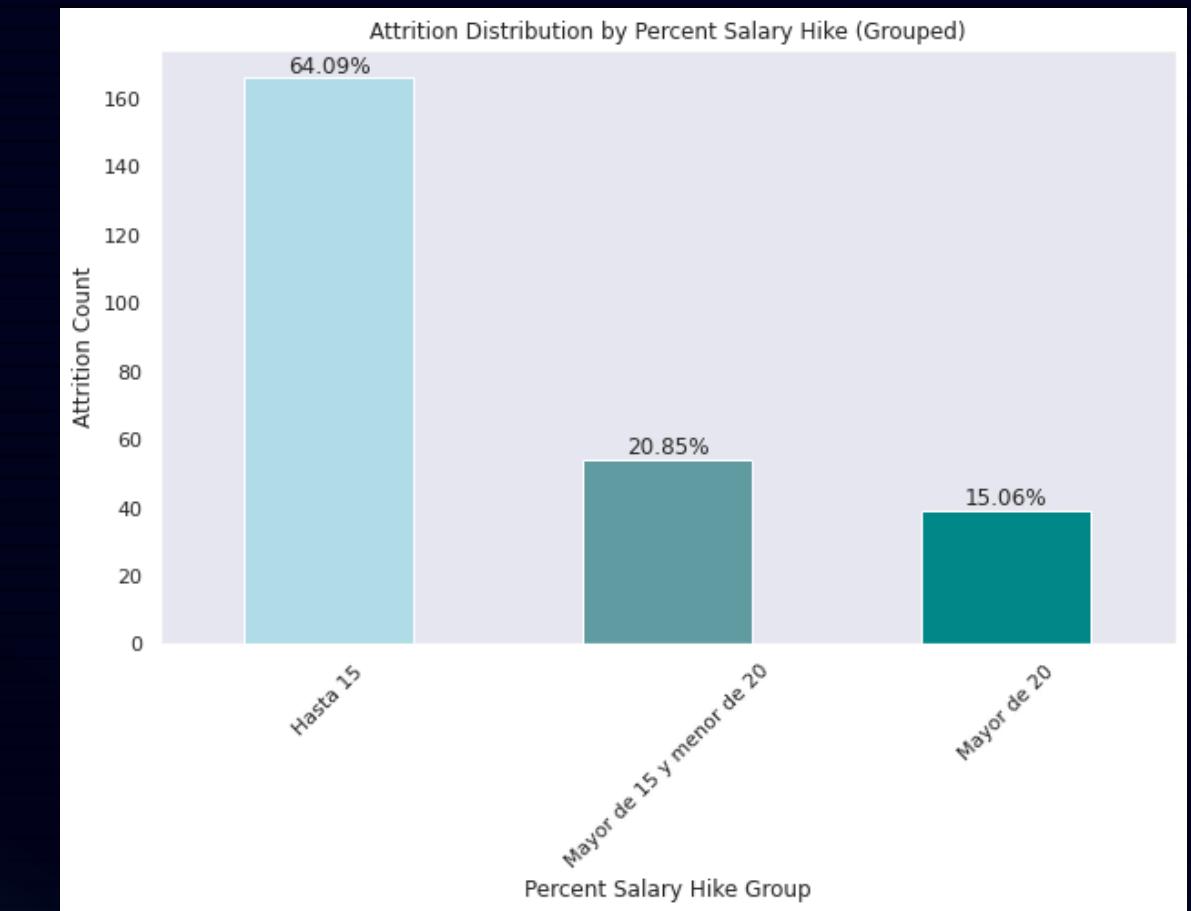
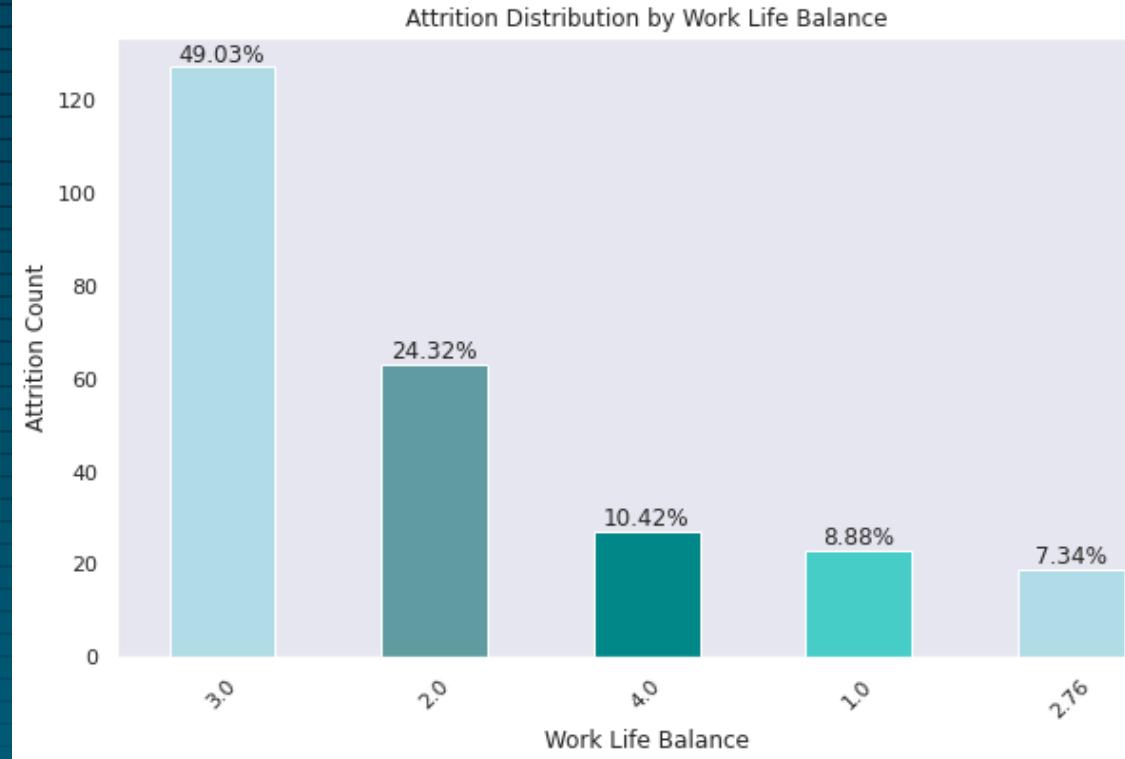
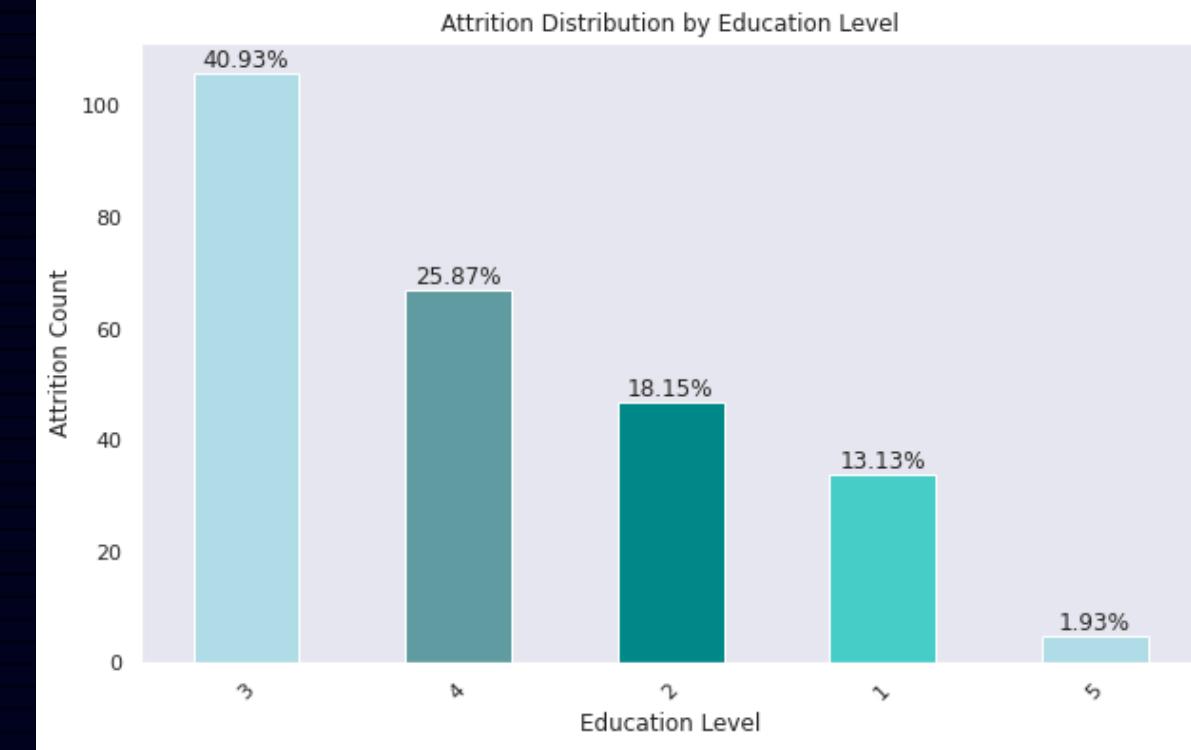
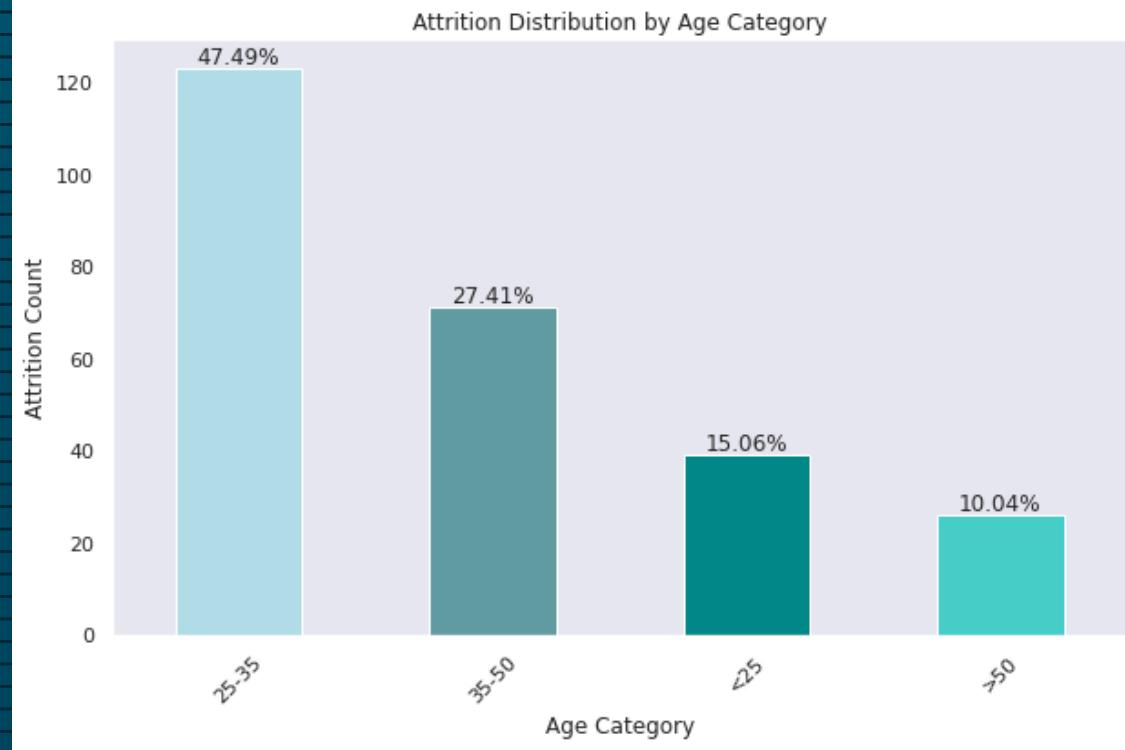
Relaciones menos relevantes

- Distancia al trabajo.
- Trabajo en remoto.
- Género.

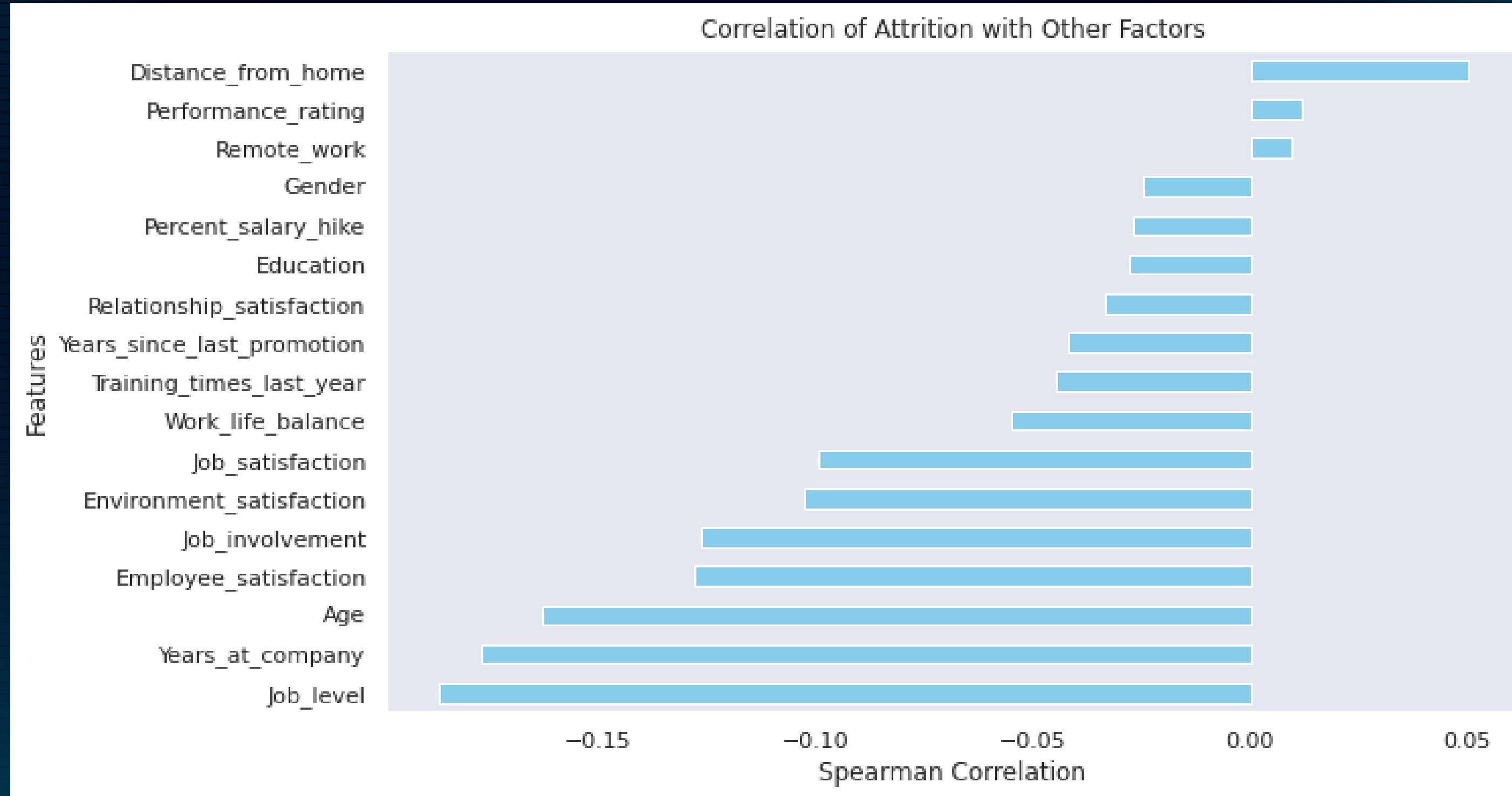
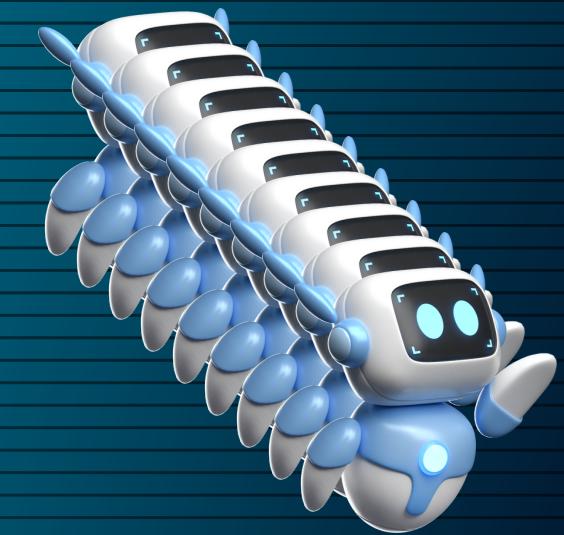




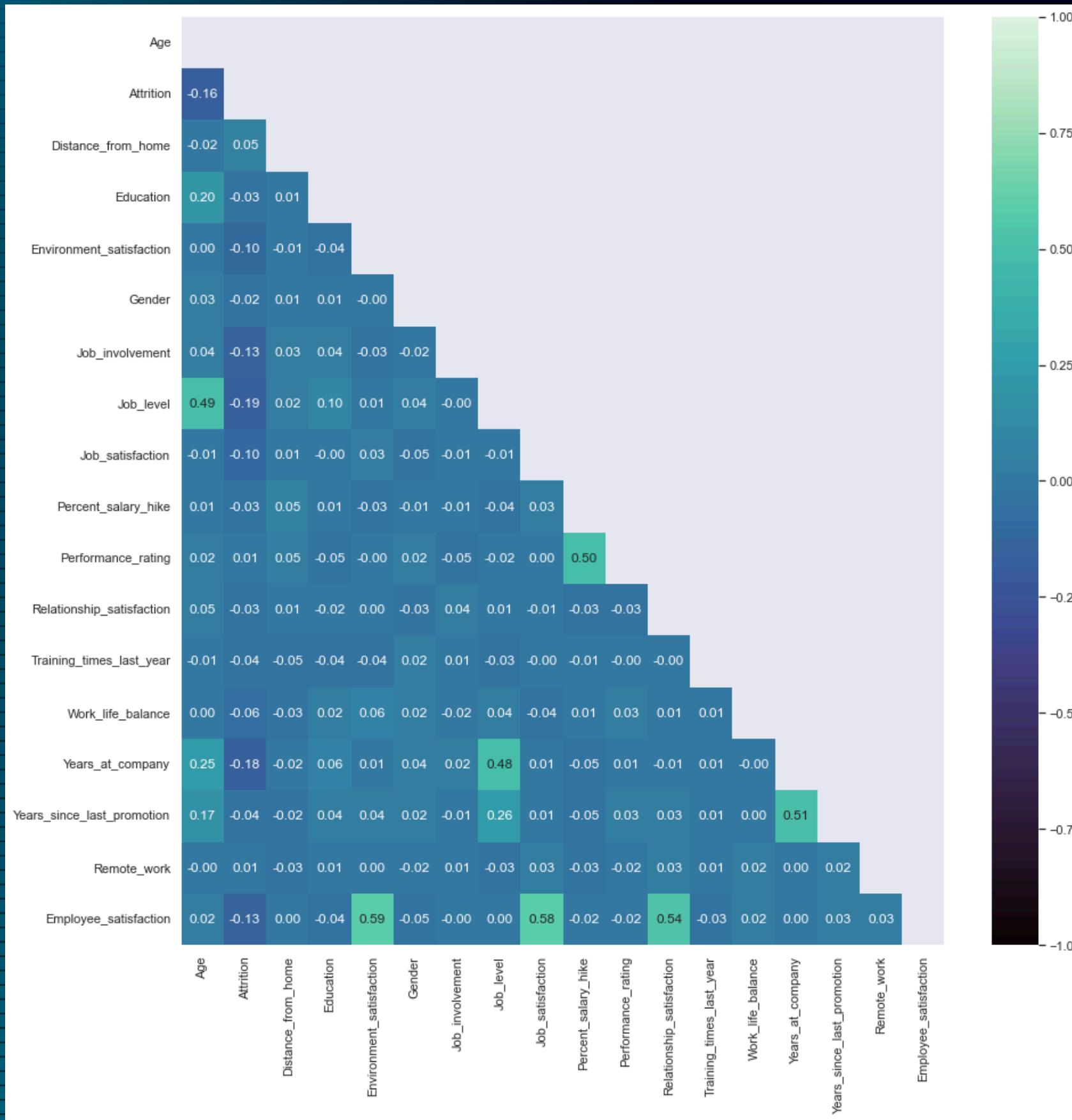
Relaciones más relevantes



Análisis de la relación entre el conjunto de características y la tasa de rotación



Análisis de correlación entre características



Conclusiones finales

- No existe una relación fuerte entre la tasa de bajas y factores laborales individuales.
- Las bajas se producen por un conjunto amplio y diverso de factores, sin predominar ninguno especialmente.
- Es el grupo de trabajadores que se encuentran en la zona media de factores como la edad, el nivel educativo o la conciliación laboral, el que tiene una mayor tendencia a dejar la empresa.
- La compañía parece mantener a los empleados relativamente satisfechos.

+ info



DATAMINDS

MUCHAS GRACIAS

