Rotación de Personal - Empresa XYZ

El área de recursos humanos de la empresa XYZ, requiere ayuda de los científicos de datos por el hecho de que se están reflejando una fuga de empleados con respecto a denuncias con cierto perfil de empleados, pero no sabe con seguridad que tipo de empleados y con qué condiciones estos están renunciando, por lo tanto, requiere la ayuda del área de datos para aclarar la situación y proponer estrategias al respecto.

IMPORTACION DE LIBRERIA

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
df = pd.read_csv('AbandonoEmpleados.csv', sep = ';', index_col= 'id', na_values='#N/D')
```

In [3]:
```
df
```

Out[3]:

id	edad	abandono	viajes	departamento	distancia_casa	educacion	carrera	empleados	satisfaccion_entorno	sexo	...	sat
1	41	Yes	Travel_Rarely	Sales	1	Universitaria	Life Sciences	1	Media	3.0	...	
2	49	No	Travel_Frequently	Research & Development	8	Secundaria	Life Sciences	1	Alta	2.0	...	
4	37	Yes	Travel_Rarely	Research & Development	2	Secundaria	Other	1	Muy_Alta	2.0	...	
5	33	No	Travel_Frequently	Research & Development	3	Universitaria	Life Sciences	1	Muy_Alta	3.0	...	
7	27	No	Travel_Rarely	Research & Development	2	Universitaria	Medical	1	Baja	3.0	...	
...	...	...	...	...	...	...	...	...	...	...	...	...
2061	36	No	Travel_Frequently	Research & Development	23	Master	Medical	1	Alta	4.0	...	
2062	39	No	Travel_Rarely	Research & Development	6	Secundaria	Medical	1	Muy_Alta	2.0	...	
2064	27	No	Travel_Rarely	Research & Development	4	Master	Life Sciences	1	Media	4.0	...	
2065	49	No	Travel_Frequently	Sales	2	Secundaria	Medical	1	Muy_Alta	NaN	...	
2068	34	No	Travel_Rarely	Research & Development	8	NaN	Medical	1	Media	4.0	...	

1470 rows × 31 columns

PROCESANDO LOS DATOS

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1470 entries, 1 to 2068
Data columns (total 31 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   edad                      1470 non-null   int64
 1   abandono                  1470 non-null   object
 2   viajes                    1470 non-null   object
 3   departamento              1470 non-null   object
 4   distancia_casa            1470 non-null   int64
 5   educacion                 1369 non-null   object
 6   carrera                   1470 non-null   object
 7   empleados                 1470 non-null   int64
 8   satisfaccion_entorno      1470 non-null   object
 9   sexo                      1271 non-null   float64
 10  implicacion               1452 non-null   object
 11  nivel_laboral             1470 non-null   int64
 12  puesto                    1470 non-null   object
 13  satisfaccion_trabajo      1394 non-null   object
 14  estado_civil              1470 non-null   object
 15  salario_mes               1470 non-null   int64
 16  num_empresas_anteriores   1470 non-null   int64
 17  mayor_edad                1470 non-null   object
 18  horas_extra               1470 non-null   object
 19  incremento_salario_porc   1470 non-null   int64
 20  evaluacion                1470 non-null   object
 21  satisfaccion_companeros   1470 non-null   object
 22  horas_quincena            1470 non-null   int64
 23  nivel_acciones            1470 non-null   int64
 24  anos_experiencia          1470 non-null   int64
 25  num_formaciones_ult_ano   1470 non-null   int64
 26  conciliacion              459 non-null    object
 27  anos_compania             1470 non-null   int64
 28  anos_en_puesto            232 non-null    float64
 29  anos_desde_ult_promocion  1470 non-null   int64
 30  anos_con_manager_actual   1470 non-null   int64
dtypes: float64(2), int64(14), object(15)
memory usage: 367.5+ KB
```

In [6]:
```
df.isna().sum().sort_values(ascending = False)
```

Out[6]:
```
anos_en_puesto              1238
conciliacion                1011
sexo                         199
educacion                    101
satisfaccion_trabajo          76
implicacion                   18
edad                           0
nivel_acciones                 0
evaluacion                     0
satisfaccion_companeros        0
horas_quincena                 0
anos_experiencia               0
horas_extra                    0
num_formaciones_ult_ano        0
anos_compania                  0
anos_desde_ult_promocion       0
incremento_salario_porc        0
salario_mes                    0
mayor_edad                     0
num_empresas_anteriores        0
abandono                       0
estado_civil                   0
puesto                         0
nivel_laboral                  0
satisfaccion_entorno           0
empleados                      0
carrera                        0
distancia_casa                 0
departamento                   0
viajes                         0
anos_con_manager_actual        0
dtype: int64
```

Conclusiones:

- anos_en_puesto y conciliacion tienen demasiados nulos --> eliminar Variables
- sexo, educacion, satisfaccion_trabajo e implicacion --> imputarlos tras EDA

In [8]:
```
df.drop(columns = ['anos_en_puesto','conciliacion'], inplace = True)
df
```

Out[8]:

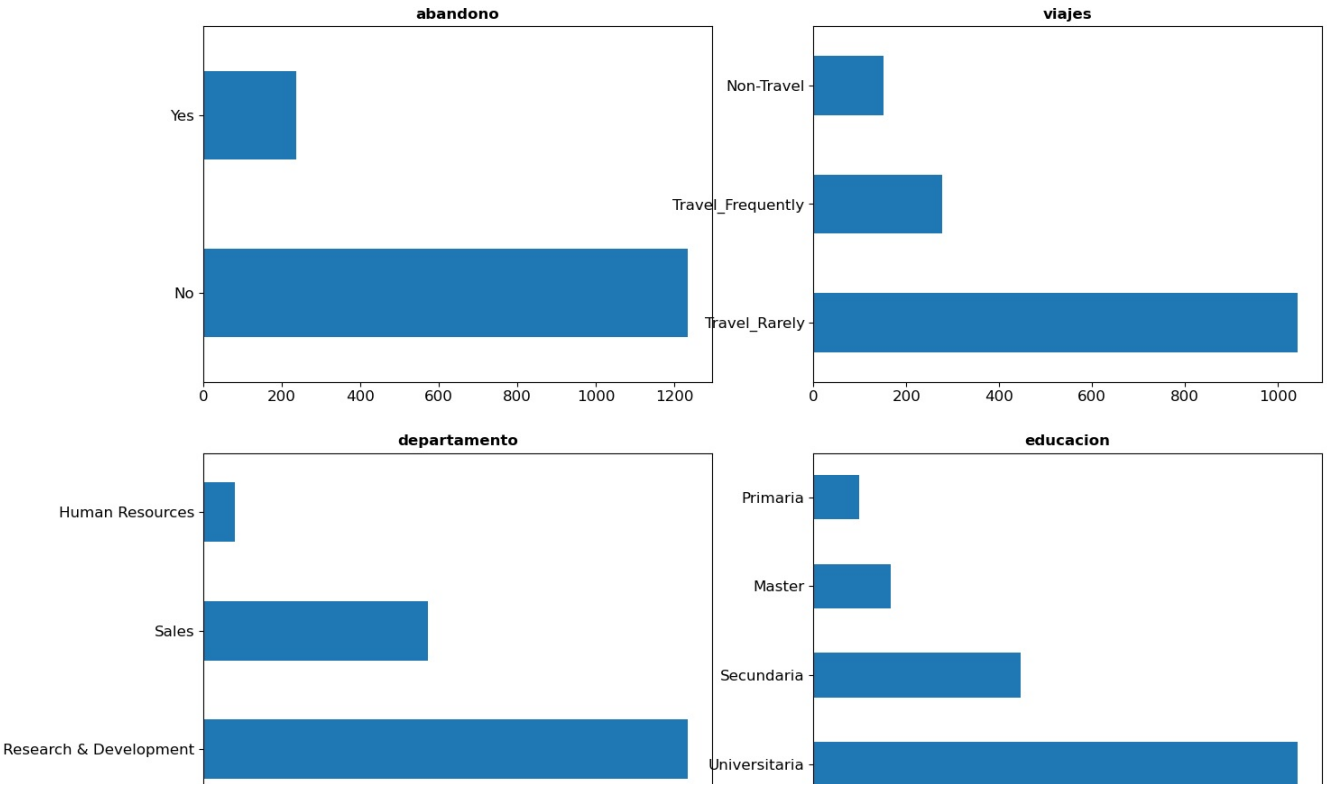
id	edad	abandono	viajes	departamento	distancia_casa	educacion	carrera	empleados	satisfaccion_entorno	sexo	...	inc
1	41	Yes	Travel_Rarely	Sales	1	Universitaria	Life Sciences	1	Media	3.0	...	
2	49	No	Travel_Frequently	Research & Development	8	Secundaria	Life Sciences	1	Alta	2.0	...	
4	37	Yes	Travel_Rarely	Research & Development	2	Secundaria	Other	1	Muy_Alta	2.0	...	
5	33	No	Travel_Frequently	Research & Development	3	Universitaria	Life Sciences	1	Muy_Alta	3.0	...	
7	27	No	Travel_Rarely	Research & Development	2	Universitaria	Medical	1	Baja	3.0	...	
...	...	...	...	...	...	...	...	...	...	...	...	...
2061	36	No	Travel_Frequently	Research & Development	23	Master	Medical	1	Alta	4.0	...	
2062	39	No	Travel_Rarely	Research & Development	6	Secundaria	Medical	1	Muy_Alta	2.0	...	
2064	27	No	Travel_Rarely	Research & Development	4	Master	Life Sciences	1	Media	4.0	...	
2065	49	No	Travel_Frequently	Sales	2	Secundaria	Medical	1	Muy_Alta	NaN	...	
2068	34	No	Travel_Rarely	Research & Development	8	NaN	Medical	1	Media	4.0	...	

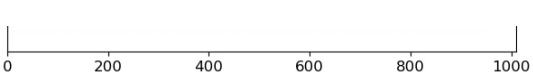
1470 rows × 29 columns

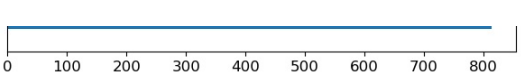
EDA VARIABLES CATEGÓRICAS

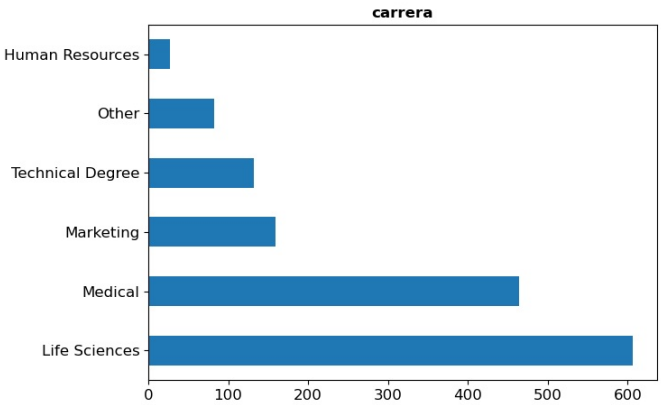
In [11]:
```
def graficos_eda_categoricos(cat):

    #Calculamos el número de filas que necesitamos
    from math import ceil
    filas = ceil(cat.shape[1] / 2)

    #Definimos el gráfico
    f, ax = plt.subplots(nrows = filas, ncols = 2, figsize = (16, filas * 6))

    #Aplanamos para iterar por el gráfico como si fuera de 1 dimensión en lugar de 2
    ax = ax.flat

    #Creamos el bucle que va añadiendo gráficos
    for cada, variable in enumerate(cat):
        cat[variable].value_counts().plot.barh(ax = ax[cada])
        ax[cada].set_title(variable, fontsize = 12, fontweight = "bold")
        ax[cada].tick_params(labelsize = 12)
```

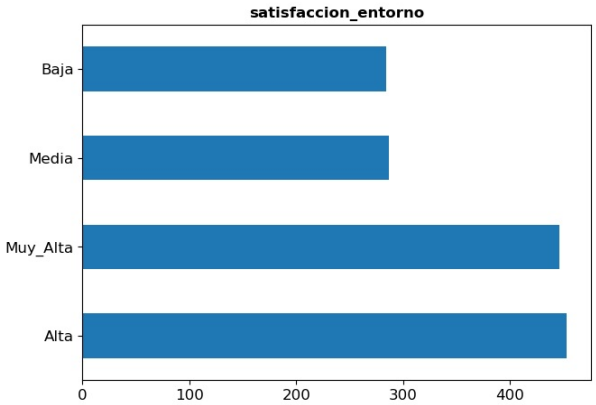
In [12]:
```
graficos_eda_categoricos(df.select_dtypes('O'))
```

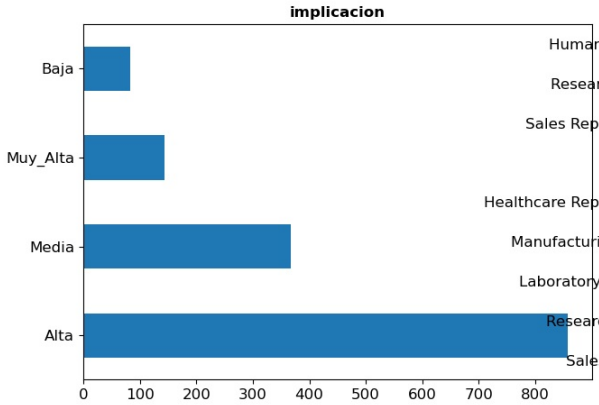


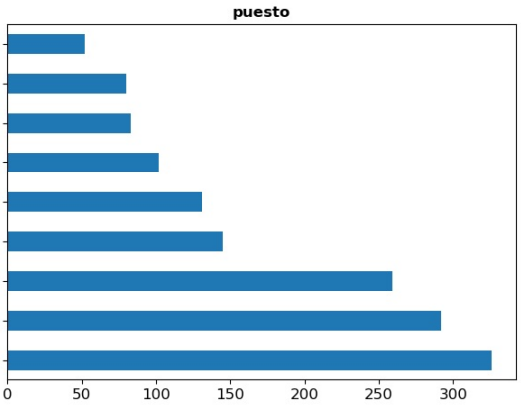

0
200
400
600
800
1000

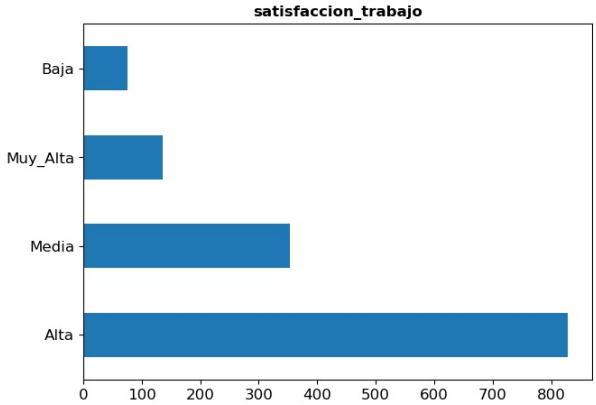

0
100
200
300
400
500
600
700
800

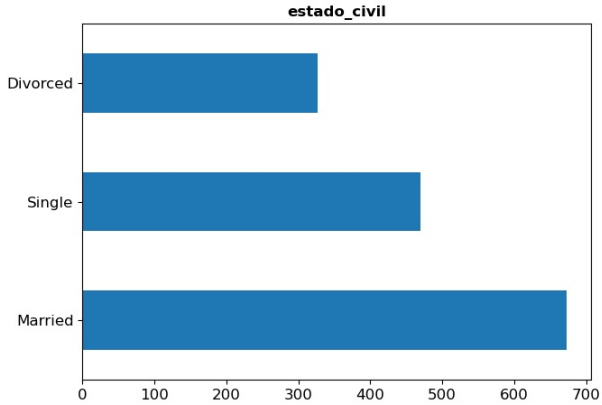

carrera
Human Resources
Other
Technical Degree
Marketing
Medical
Life Sciences
0
100
200
300
400
500
600

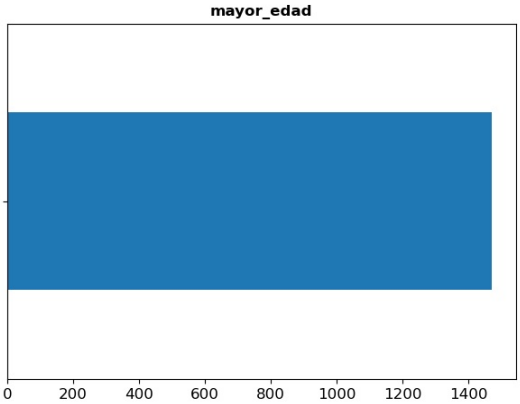

satisfaccion_entorno
Baja
Media
Muy_Alta
Alta
0
100
200
300
400

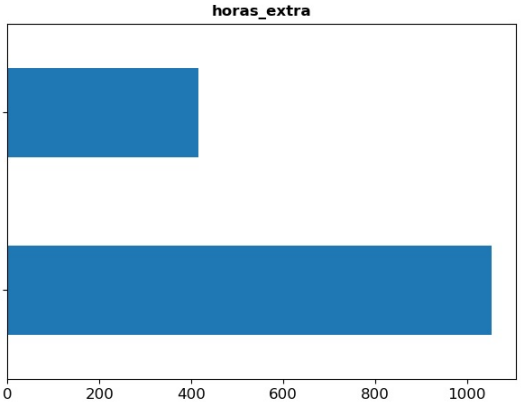

implicacion
Baja
Muy_Alta
Media
Alta
0
100
200
300
400
500
600
700
800

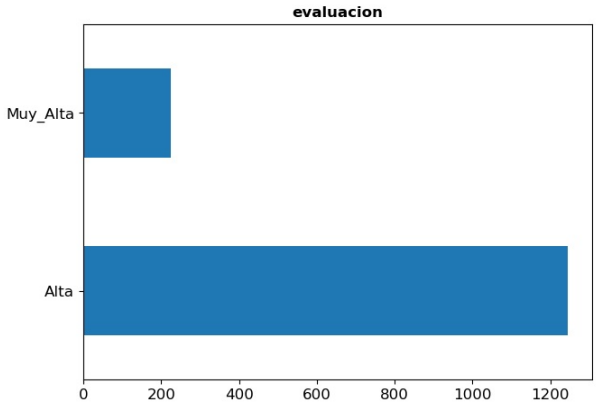

puesto
Human Resources
Research Director
Sales Representative
Manager
Healthcare Representative
Manufacturing Director
Laboratory Technician
Research Scientist
Sales Executive
0
50
100
150
200
250
300

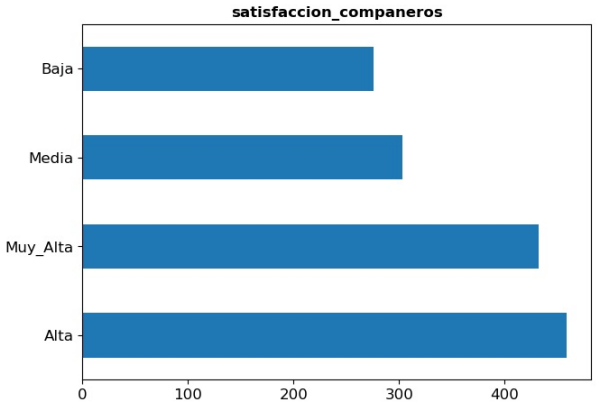

satisfaccion_trabajo
Baja
Muy_Alta
Media
Alta
0
100
200
300
400
500
600
700
800


estado_civil
Divorced
Single
Married
0
100
200
300
400
500
600
700


mayor_edad
Y
0
200
400
600
800
1000
1200
1400


horas_extra
Yes
No
0
200
400
600
800
1000


evaluacion
Muy_Alta
Alta
0
200
400
600
800
1000
1200


satisfaccion_companeros
Baja
Media
Muy_Alta
Alta
0
100
200
300
400

Conclusiones:

- mayor_edad solo tiene un valor --> eliminarla
- Sobre las imputaciones pendientes de variables categóricas:
  - educacion: imputar por 'Universitaria'
  - satisfaccion_trabajo: imputar por 'Alta'
  - implicacion: imputar por 'Alta'

EDA VARIABLES NUMÉRICAS

In [13]:
```
def estadisticos_cont(num):
    #Calculamos describe
    estadisticos = num.describe().T
    #Añadimos la mediana
    estadisticos['median'] = num.median()
    #Reordenamos para que la mediana esté al lado de la media
    estadisticos = estadisticos.iloc[:,[0,1,8,2,3,4,5,6,7]]
    #Lo devolvemos
    return(estadisticos)
```

In [14]:
```
estadisticos_cont(df.select_dtypes('number'))
```

Out[14]:

	count	mean	median	std	min	25%	50%	75%	max
edad	1470.0	36.923810	36.0	9.135373	18.0	30.0	36.0	43.0	60.0
distancia_casa	1470.0	9.192517	7.0	8.106864	1.0	2.0	7.0	14.0	29.0
empleados	1470.0	1.000000	1.0	0.000000	1.0	1.0	1.0	1.0	1.0
sexo	1271.0	2.727773	3.0	0.720788	1.0	2.0	3.0	3.0	4.0
nivel_laboral	1470.0	2.063946	2.0	1.106940	1.0	1.0	2.0	3.0	5.0
salario_mes	1470.0	6502.931293	4919.0	4707.956783	1009.0	2911.0	4919.0	8379.0	19999.0
num_empresas_anteriores	1470.0	2.693197	2.0	2.498009	0.0	1.0	2.0	4.0	9.0
incremento_salario_porc	1470.0	15.209524	14.0	3.659938	11.0	12.0	14.0	18.0	25.0
horas_quincena	1470.0	80.000000	80.0	0.000000	80.0	80.0	80.0	80.0	80.0
nivel_acciones	1470.0	0.793878	1.0	0.852077	0.0	0.0	1.0	1.0	3.0
anos_experiencia	1470.0	11.279592	10.0	7.780782	0.0	6.0	10.0	15.0	40.0
num_formaciones_ult_ano	1470.0	2.799320	3.0	1.289271	0.0	2.0	3.0	3.0	6.0
anos_compania	1470.0	7.008163	5.0	6.126525	0.0	3.0	5.0	9.0	40.0
anos_desde_ult_promocion	1470.0	2.187755	1.0	3.222430	0.0	0.0	1.0	3.0	15.0
anos_con_manager_actual	1470.0	4.123129	3.0	3.568136	0.0	2.0	3.0	7.0	17.0

Conclusiones:

- Empleados solo tiene un valor --> Eliminarla
- Sexo tiene 4 valores --> Eliminarla
- Horas quincena solo tiene una valor --> Eliminarla
- De los nulos pendientes de imputación que sean numéricas solo está el sexo, pero como la vamos a eliminar ya no hay que imputar nada

In [16]:
```
df.drop(columns = ['empleados','sexo','horas_quincena'], inplace = True)
df
```

Out[16]:

id	edad	abandono	viajes	departamento	distancia_casa	educacion	carrera	satisfaccion_entorno	implicacion	nivel_laboral
1	41	Yes	Travel_Rarely	Sales	1	Universitaria	Life Sciences	Media	Alta	2
2	49	No	Travel_Frequently	Research & Development	8	Secundaria	Life Sciences	Alta	Media	2
4	37	Yes	Travel_Rarely	Research & Development	2	Secundaria	Other	Muy_Alta	Media	1
5	33	No	Travel_Frequently	Research & Development	3	Universitaria	Life Sciences	Muy_Alta	Alta	1
7	27	No	Travel_Rarely	Research & Development	2	Universitaria	Medical	Baja	Alta	1
...	...	...	...	...	...	...	...	...	...	...
2061	36	No	Travel_Frequently	Research & Development	23	Master	Medical	Alta	Muy_Alta	2
2062	39	No	Travel_Rarely	Research & Development	6	Secundaria	Medical	Muy_Alta	Media	3
2064	27	No	Travel_Rarely	Research & Development	4	Master	Life Sciences	Media	Muy_Alta	2
2065	49	No	Travel_Frequently	Sales	2	Secundaria	Medical	Muy_Alta	Media	2
2068	34	No	Travel_Rarely	Research & Development	8	NaN	Medical	Media	Muy_Alta	2

1470 rows × 26 columns

GENERACIÓN DE INSIGHTS

Cuantificación del problema: ¿Cual es la tasa de abandono?

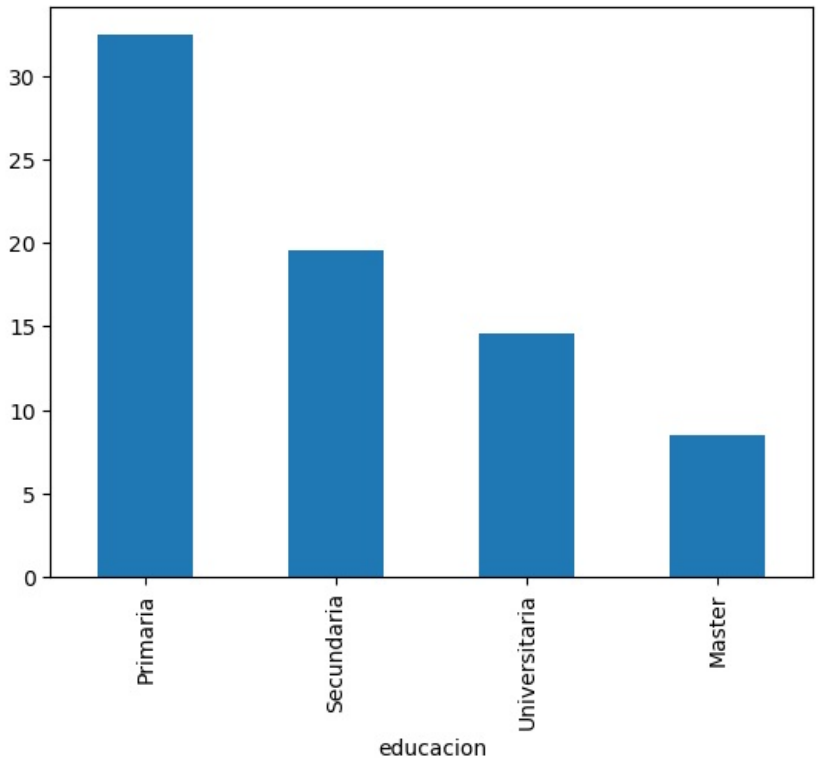
In [22]:
```
df.abandono.value_counts(normalize = True) *100
```

Out[22]:
```
No     83.877551
Yes    16.122449
Name: abandono, dtype: float64
```

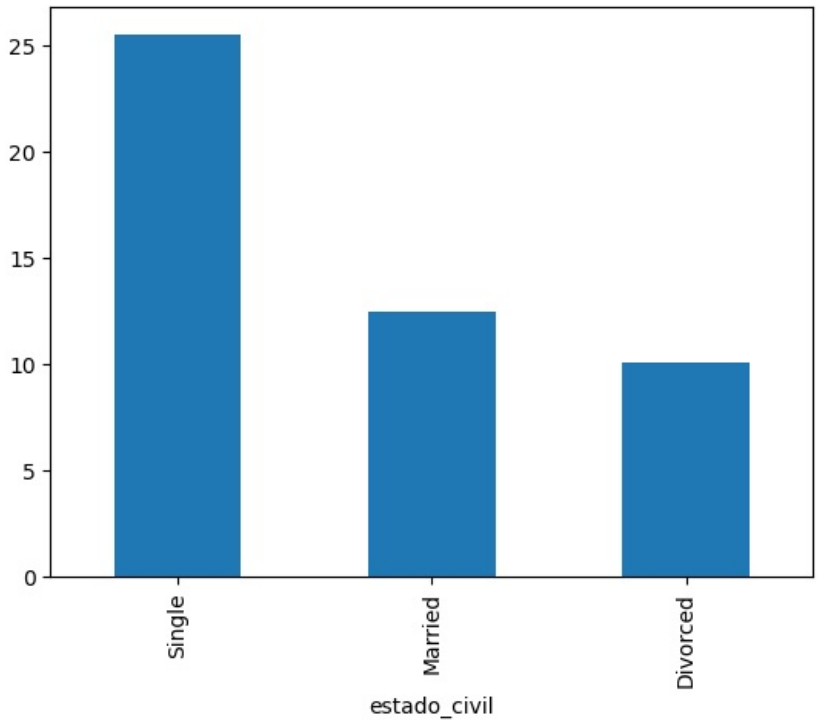
¿Hay un perfil tipo de empleado que deja la empresa?

In [40]:
```
# Transformar abandono a numérica
df['abandono'] = df.abandono.map({'No':0, 'Yes':1})
```

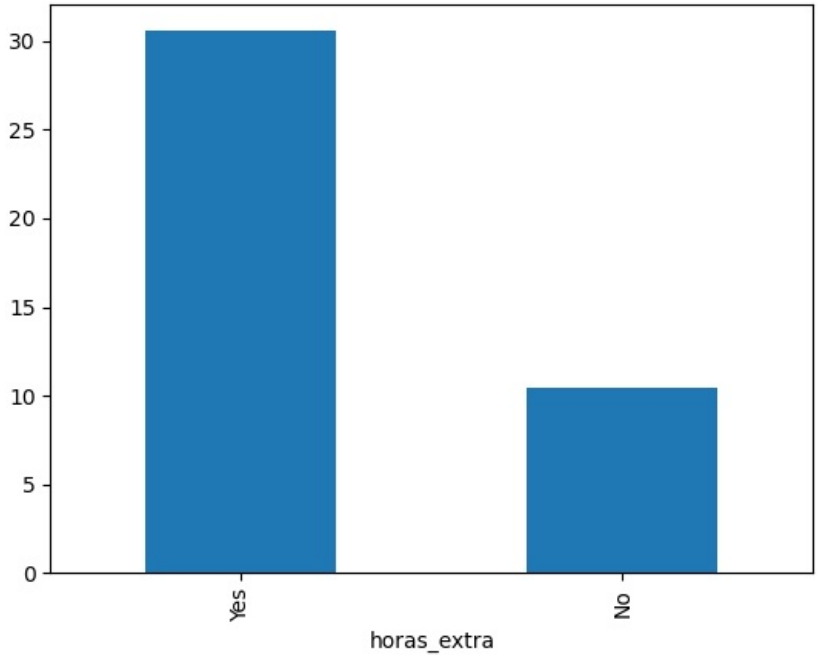
In [16]:
```
# Analisis por educación
temp = df.groupby('educacion').abandono.mean().sort_values(ascending = False) * 100
temp.plot.bar();
```



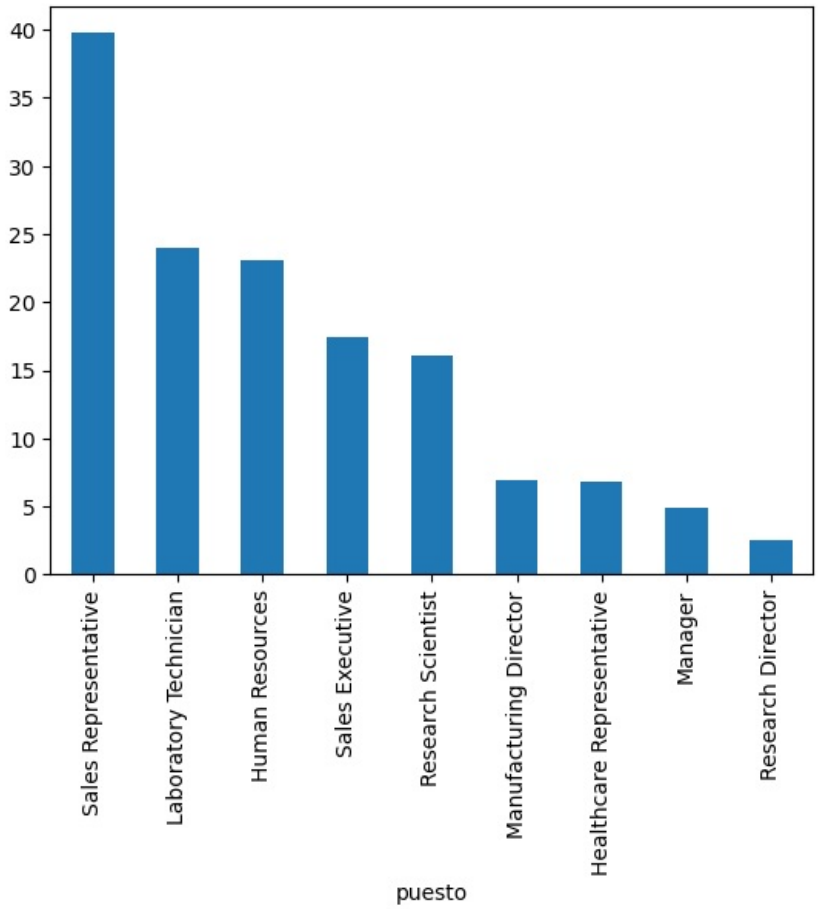
In [17]:
```
# Analisis por estado civil
temp = df.groupby('estado_civil').abandono.mean().sort_values(ascending = False) * 100
temp.plot.bar();
```



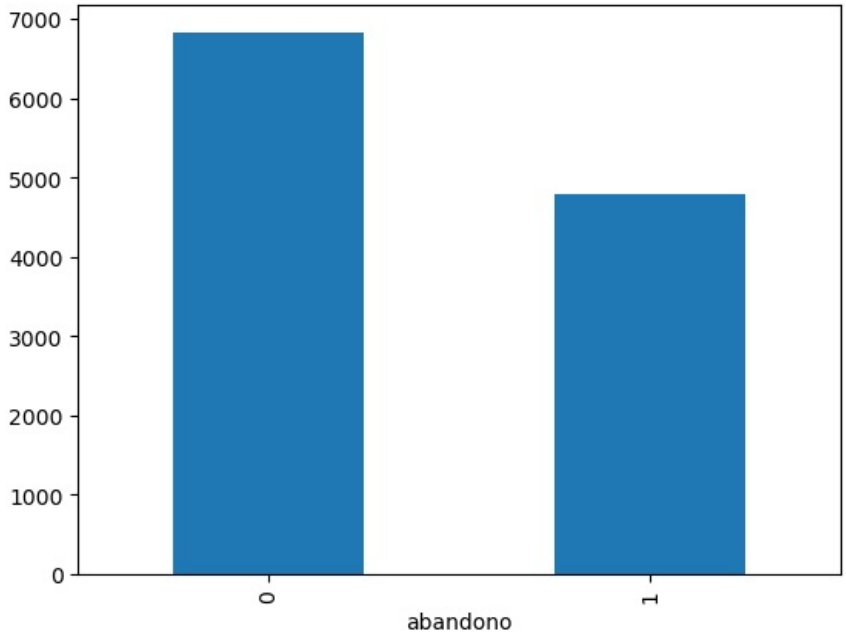
In [18]:
```
# Analisis por horas extras
temp = df.groupby('horas_extra').abandono.mean().sort_values(ascending = False) * 100
temp.plot.bar();
```



In [19]:
```
# Analisis por puesto
temp = df.groupby('puesto').abandono.mean().sort_values(ascending = False) * 100
temp.plot.bar();
```



In [20]:
```
temp = df.groupby('abandono').salario_mes.mean()
temp.plot.bar();
```



Conclusiones:

El perfil medio del empleado que deja la empresa es:

- Bajo nivel educativo
- Soltero
- Trabaja en ventas
- Bajo salario
- Alta carga de horas extras

¿Cual es el impacto económico de este problema?

¿Cuál es el impacto económico de este problema?

Según el estudio "Cost of Turnover" del Center for American Progress:

- El coste de la fuga de los empleados que ganan menos de 30000 es del 16,1% de su salario
- El coste de la fuga de los empleados que ganan entre 30000-50000 es del 19,7% de su salario
- El coste de la fuga de los empleados que ganan entre 50000-75000 es del 20,4% de su salario
- El coste de la fuga de los empleados que ganan más de 75000 es del 21% de su salario

In [21]:
```
# Creamos una nueva variable salario_ano del empleado
df['salario_ano'] = df.salario_mes.transform(lambda x: x*12)
df[['salario_mes','salario_ano']]
```

Out[21]:

id	salario_mes	salario_ano
1	5993	71916
2	5130	61560
4	2090	25080
5	2909	34908
7	3468	41616
...	...	...
2061	2571	30852
2062	9991	119892
2064	6142	73704
2065	5390	64680
2068	4404	52848

1470 rows × 2 columns

In [22]:
```
# Calculamos el impacto económico de cada empleado si deja la empresa

#Lista de condiciones
condiciones = [(df['salario_ano'] <= 30000),
               (df['salario_ano'] > 30000) & (df['salario_ano'] <= 50000),
               (df['salario_ano'] > 50000) & (df['salario_ano'] <= 75000),
               (df['salario_ano'] > 75000)]

#Lista de resultados
resultados = [df.salario_ano * 0.161, df.salario_ano * 0.197, df.salario_ano * 0.204, df.salario_ano * 0.21]

#Aplicamos select
df['impacto_abandono'] = np.select(condiciones,resultados, default = -999)

df
```

Out[22]:

id	edad	abandono	viajes	departamento	distancia_casa	educacion	carrera	satisfaccion_entorno	implicacion	nivel_laboral
1	41	1	Travel_Rarely	Sales	1	Universitaria	Life Sciences	Media	Alta	2
2	49	0	Travel_Frequently	Research & Development	8	Secundaria	Life Sciences	Alta	Media	2
4	37	1	Travel_Rarely	Research & Development	2	Secundaria	Other	Muy_Alta	Media	1
5	33	0	Travel_Frequently	Research & Development	3	Universitaria	Life Sciences	Muy_Alta	Alta	1
7	27	0	Travel_Rarely	Research & Development	2	Universitaria	Medical	Baja	Alta	1
...	...	...	...	...	...	...	...	...	...	...
2061	36	0	Travel_Frequently	Research & Development	23	Master	Medical	Alta	Muy_Alta	2
2062	39	0	Travel_Rarely	Research & Development	6	Secundaria	Medical	Muy_Alta	Media	3
2064	27	0	Travel_Rarely	Research & Development	4	Master	Life Sciences	Media	Muy_Alta	2
2065	49	0	Travel_Frequently	Sales	2	Secundaria	Medical	Muy_Alta	Media	2
2068	34	0	Travel_Rarely	Research & Development	8	Universitaria	Medical	Media	Muy_Alta	2

1470 rows × 27 columns

¿Cúanto nos ha costado este problema en el último año?

In [23]:
```
coste_total =  df.loc[df.abandono == 1].impacto_abandono.sum()
coste_total
```

Out[23]:
```
2719005.912
```

¿Cuanto nos cuesta que los empleados no estén motivados? (pérdidas en implicación == Baja)

In [24]:
```
df.loc[(df.abandono == 1) & (df.implicacion == 'Baja')].impacto_abandono.sum()
```

Out[24]:
```
368672.688
```

¿Cuanto dinero podríamos ahorrar fidelizando mejor a nuestros empleados?

In [25]:
```
print(f"Reducir un 10% la fuga de empleados nos ahorraría {int(coste_total * 0.1)}$ cada año.")

print(f"Reducir un 20% la fuga de empleados nos ahorraría {int(coste_total * 0.2)}$ cada año.")

print(f"Reducir un 30% la fuga de empleados nos ahorraría {int(coste_total * 0.3)}$ cada año.")
```

```
Reducir un 10% la fuga de empleados nos ahorraría 271900$ cada año.
Reducir un 20% la fuga de empleados nos ahorraría 543801$ cada año.
Reducir un 30% la fuga de empleados nos ahorraría 815701$ cada año.
```

Y podemos seguir trazando estrategias asociadas a los insights de abandono:

Habíamos visto que los representantes de ventas son el puesto que más se van. ¿Tendría sentido hacer un plan específico para ellos? ¿Cual sería el coste ahorrado si disminuimos la fuga un 30%?

Primero vamos a calcular el % de representantes de ventas que se han ido el año pasado

In [26]:
```
total_repre_pasado = len(df.loc[df.puesto == 'Sales Representative'])
abandonos_repre_pasado = len(df.loc[(df.puesto == 'Sales Representative') & (df.abandono == 1)])
porc_pasado = abandonos_repre_pasado / total_repre_pasado

porc_pasado
```

Out[26]:
```
0.39759036144578314
```

Ahora vamos a estimar cuántos se nos irán este año

In [27]:
```
total_repre_actual = len(df.loc[(df.puesto == 'Sales Representative') & (df.abandono == 0)])
se_iran = int(total_repre_actual * porc_pasado)

se_iran
```

Out[27]:
```
19
```

Sobre ellos cuantos podemos retener (hipótesis 30%) y cuanto dinero puede suponer

In [28]:
```
retenemos = int(se_iran * 0.3)

ahorramos = df.loc[(df.puesto == 'Sales Representative') & (df.abandono == 0),'impacto_abandono'].sum() * porc_

print(f'Podemos retener {retenemos} representantes de ventas y ello supondría ahorrar {ahorramos}$.')
```

```
Podemos retener 5 representantes de ventas y ello supondría ahorrar 37447.22424578312$.
```

Este dato también es muy interesante porque nos permite determinar el presupuesto para acciones de retención por departamento o perfil.

Ya que sabemos que podemos gastarnos hasta 37.000$ sólo en acciones específicas para retener a representantes de ventas y se estarían pagando sólas con la pérdida evitada

DIA 3: MODELO DE MACHINE LEARNING

In [29]:
```
df_ml = df.copy()
```

In [30]:
```
df_ml.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1470 entries, 1 to 2068
Data columns (total 27 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   edad                      1470 non-null   int64
 1   abandono                  1470 non-null   int64
 2   viajes                    1470 non-null   object
 3   departamento              1470 non-null   object
 4   distancia_casa            1470 non-null   int64
 5   educacion                 1470 non-null   object
 6   carrera                   1470 non-null   object
 7   satisfaccion_entorno      1470 non-null   object
 8   implicacion               1470 non-null   object
 9   nivel_laboral             1470 non-null   int64
 10  puesto                    1470 non-null   object
 11  satisfaccion_trabajo      1470 non-null   object
 12  estado_civil              1470 non-null   object
 13  salario_mes               1470 non-null   int64
 14  num_empresas_anteriores   1470 non-null   int64
 15  horas_extra               1470 non-null   object
 16  incremento_salario_porc   1470 non-null   int64
 17  evaluacion                1470 non-null   object
 18  satisfaccion_companeros   1470 non-null   object
 19  nivel_acciones            1470 non-null   int64
 20  anos_experiencia          1470 non-null   int64
 21  num_formaciones_ult_ano   1470 non-null   int64
 22  anos_compania             1470 non-null   int64
 23  anos_desde_ult_promocion  1470 non-null   int64
 24  anos_con_manager_actual   1470 non-null   int64
 25  salario_ano               1470 non-null   int64
 26  impacto_abandono          1470 non-null   float64
dtypes: float64(1), int64(14), object(12)
memory usage: 321.6+ KB
```

PREPARACIÓN DE LOS DATOS PARA LA MODELIZACIÓN

Transformar todas las variables categóricas a numéricas

Transformar todas las variables categóricas a numéricas

In [31]:
```
from sklearn.preprocessing import OneHotEncoder

#Categóricas
cat = df_ml.select_dtypes('O')

#Instanciamos
ohe = OneHotEncoder(sparse = False)

#Entrenamos
ohe.fit(cat)

#Aplicamos
cat_ohe = ohe.transform(cat)

#Ponemos los nombres
cat_ohe = pd.DataFrame(cat_ohe, columns = ohe.get_feature_names_out(input_features = cat.columns)).reset_index(
```

```
C:\Users\ASUS\anaconda3\lib\site-packages\sklearn\preprocessing\_encoders.py:828: FutureWarning: `sparse` was r
enamed to `sparse_output` in version 1.2 and will be removed in 1.4. `sparse_output` is ignored unless you leav
e `sparse` to its default value.
  warnings.warn(
```

In [32]:
```
cat_ohe
```

Out[32]:

	viajes_Non-Travel	viajes_Travel_Frequently	viajes_Travel_Rarely	departamento_Human Resources	departamento_Research & Development	departamento_Sales	educaci
0	0.0	0.0	1.0	0.0	0.0	1.0	
1	0.0	1.0	0.0	0.0	1.0	0.0	
2	0.0	0.0	1.0	0.0	1.0	0.0	
3	0.0	1.0	0.0	0.0	1.0	0.0	
4	0.0	0.0	1.0	0.0	1.0	0.0	
...	...	...	...	...	...	...	
1465	0.0	1.0	0.0	0.0	1.0	0.0	
1466	0.0	0.0	1.0	0.0	1.0	0.0	
1467	0.0	0.0	1.0	0.0	1.0	0.0	
1468	0.0	1.0	0.0	0.0	0.0	1.0	
1469	0.0	0.0	1.0	0.0	1.0	0.0	

1470 rows × 48 columns

Dataframe final

Seleccionamos las variables numéricas para poder juntarlas a las cat_hoe

In [33]:
```
num = df.select_dtypes('number').reset_index(drop = True)
```

Las juntamos todas en el dataframe final

In [34]:
```
df_ml = pd.concat([cat_ohe,num], axis = 1)
df_ml
```

Out[34]:

	viajes_Non-Travel	viajes_Travel_Frequently	viajes_Travel_Rarely	departamento_Human Resources	departamento_Research & Development	departamento_Sales	educaci
0	0.0	0.0	1.0	0.0	0.0	1.0	
1	0.0	1.0	0.0	0.0	1.0	0.0	
2	0.0	0.0	1.0	0.0	1.0	0.0	
3	0.0	1.0	0.0	0.0	1.0	0.0	
4	0.0	0.0	1.0	0.0	1.0	0.0	
...	...	...	...	...	...	...	
1465	0.0	1.0	0.0	0.0	1.0	0.0	
1466	0.0	0.0	1.0	0.0	1.0	0.0	
1467	0.0	0.0	1.0	0.0	1.0	0.0	
1468	0.0	1.0	0.0	0.0	0.0	1.0	
1469	0.0	0.0	1.0	0.0	1.0	0.0	

1470 rows × 63 columns

DISEÑO DE LA MODELIZACIÓN

Separación predictoras y target

In [35]:
```
x = df_ml.drop(columns='abandono')
y = df_ml['abandono']
```

Separación train y test

In [36]:
```
from sklearn.model_selection import train_test_split

train_x, test_x, train_y, test_y = train_test_split(x, y, test_size = 0.3)
```

ENTRENAMIENTO DEL MODELO SOBRE TRAIN

In [37]:
```
from sklearn.tree import DecisionTreeClassifier

#Instanciar
ac = DecisionTreeClassifier(max_depth=4)

#Entrenar
ac.fit(train_x,train_y)
```

Out[37]:
```
DecisionTreeClassifier(max_depth=4)
```

PREDICCIÓN Y VALIDACIÓN SOBRE TEST

In [38]:
```
# Predicción
pred = ac.predict_proba(test_x)[:, 1]
pred[:20]
```

Out[38]:
```
array([0.        , 0.06105263, 0.06105263, 0.06105263, 0.14835165,
       0.14835165, 0.10606061, 0.06105263, 0.06105263, 0.06105263,
       0.11111111, 0.06105263, 0.06105263, 0.34545455, 0.06105263,
       0.5       , 0.10606061, 0.11111111, 0.06105263, 0.06105263])
```

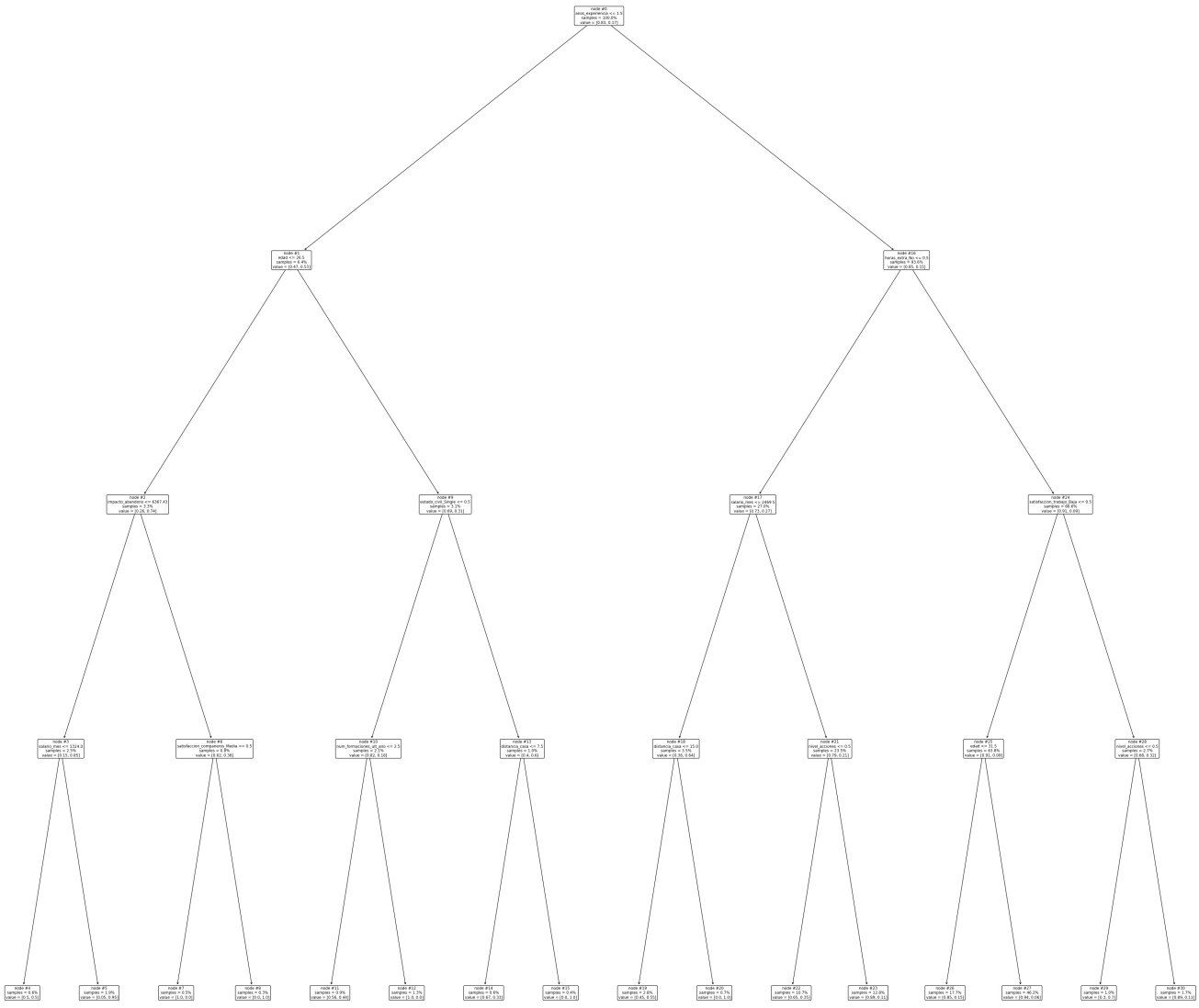
In [39]:
```
# Evaluación
from sklearn.metrics import roc_auc_score

roc_auc_score(test_y,pred)
```

Out[39]:
```
0.6596306068601583
```

INTERPRETACIÓN

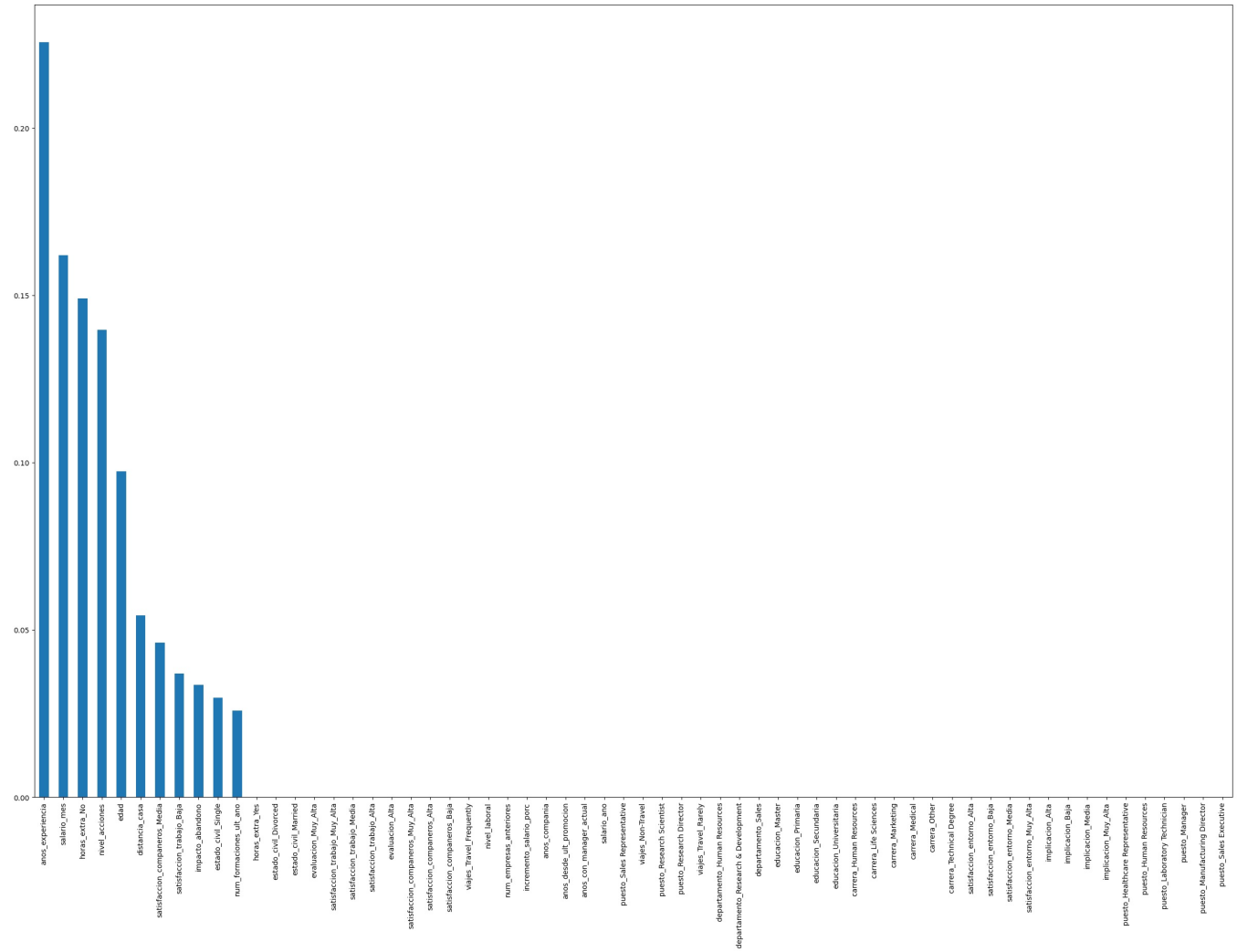
Diagrama del árbol

In [40]:
```
from sklearn.tree import plot_tree

plt.figure(figsize = (50,50))

plot_tree(ac,
          feature_names= test_x.columns,
          impurity = False,
          node_ids = True,
          proportion = True,
          rounded = True,
          precision = 2);
```



Importancia de las variables

In [41]:
```
pd.Series(ac.feature_importances_,index = test_x.columns).sort_values(ascending = False).plot(kind = 'bar', fig
```



EXPLOTACIÓN

Incoporación del scoring al dataframe principal

In [42]:
```
df['scoring_abandono'] = ac.predict_proba(df_ml.drop(columns = 'abandono'))[:, 1]
df
```

Out[42]:

id	edad	abandono	viajes	departamento	distancia_casa	educacion	carrera	satisfaccion_entorno	implicacion	nivel_laboral
1	41	1	Travel_Rarely	Sales	1	Universitaria	Life Sciences	Media	Alta	2
2	49	0	Travel_Frequently	Research & Development	8	Secundaria	Life Sciences	Alta	Media	2
4	37	1	Travel_Rarely	Research & Development	2	Secundaria	Other	Muy_Alta	Media	1
5	33	0	Travel_Frequently	Research & Development	3	Universitaria	Life Sciences	Muy_Alta	Alta	1
7	27	0	Travel_Rarely	Research & Development	2	Universitaria	Medical	Baja	Alta	1
...	...	...	...	...	...	...	...	...	...	...
2061	36	0	Travel_Frequently	Research & Development	23	Master	Medical	Alta	Muy_Alta	2
2062	39	0	Travel_Rarely	Research & Development	6	Secundaria	Medical	Muy_Alta	Media	3
2064	27	0	Travel_Rarely	Research & Development	4	Master	Life Sciences	Media	Muy_Alta	2
2065	49	0	Travel_Frequently	Sales	2	Secundaria	Medical	Muy_Alta	Media	2
2068	34	0	Travel_Rarely	Research & Development	8	Universitaria	Medical	Media	Muy_Alta	2

1470 rows × 28 columns

Ejemplo de los 10 empleados con mayor probabilidad de dejar la empresa

In [43]:
```
df.sort_values(by = 'scoring_abandono', ascending = False)[0:10]
```

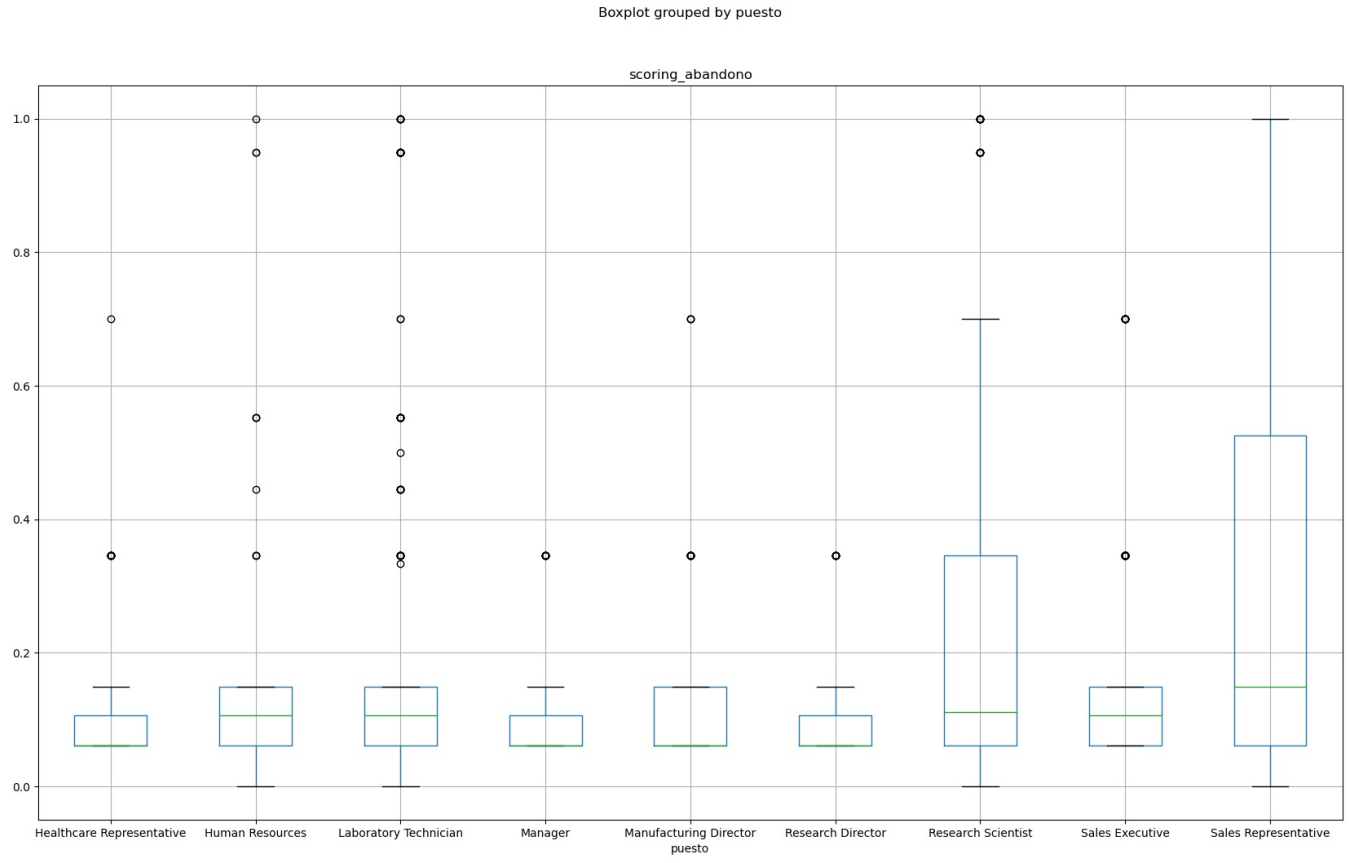
Out[43]:

id	edad	abandono	viajes	departamento	distancia_casa	educacion	carrera	satisfaccion_entorno	implicacion	nivel_laboral
1767	43	1	Travel_Frequently	Research & Development	17	Universitaria	Technical Degree	Alta	Media	1
22	22	0	Non-Travel	Research & Development	16	Master	Medical	Muy_Alta	Muy_Alta	1
911	32	1	Travel_Rarely	Research & Development	25	Universitaria	Life Sciences	Baja	Alta	1
1818	26	1	Travel_Rarely	Human Resources	20	Universitaria	Medical	Muy_Alta	Alta	1
749	29	0	Travel_Rarely	Sales	10	Universitaria	Life Sciences	Alta	Alta	1
1905	34	1	Non-Travel	Research & Development	16	Primaria	Technical Degree	Muy_Alta	Baja	1
1433	31	1	Travel_Rarely	Research & Development	8	Secundaria	Life Sciences	Baja	Media	1
881	35	1	Travel_Frequently	Research & Development	25	Universitaria	Life Sciences	Muy_Alta	Alta	1
1108	33	1	Travel_Rarely	Research & Development	25	Secundaria	Medical	Baja	Media	1
1868	29	1	Travel_Frequently	Research & Development	24	Secundaria	Life Sciences	Muy_Alta	Media	1

10 rows × 28 columns

Ejemplo: riesgo de dejar la empresa por puesto de trabajo

In [44]:
```
df.boxplot(column='scoring_abandono', by='puesto', figsize = (20,12));
```



GUARDAR EL RESULTADO

In [45]:
```
df
```

Out[45]:

id	edad	abandono	viajes	departamento	distancia_casa	educacion	carrera	satisfaccion_entorno	implicacion	nivel_laboral
1	41	1	Travel_Rarely	Sales	1	Universitaria	Life Sciences	Media	Alta	2
2	49	0	Travel_Frequently	Research & Development	8	Secundaria	Life Sciences	Alta	Media	2
4	37	1	Travel_Rarely	Research & Development	2	Secundaria	Other	Muy_Alta	Media	1
5	33	0	Travel_Frequently	Research & Development	3	Universitaria	Life Sciences	Muy_Alta	Alta	1
7	27	0	Travel_Rarely	Research & Development	2	Universitaria	Medical	Baja	Alta	1
...	...	...	...	...	...	...	...	...	...	...
2061	36	0	Travel_Frequently	Research & Development	23	Master	Medical	Alta	Muy_Alta	2
2062	39	0	Travel_Rarely	Research & Development	6	Secundaria	Medical	Muy_Alta	Media	3
2064	27	0	Travel_Rarely	Research & Development	4	Master	Life Sciences	Media	Muy_Alta	2
2065	49	0	Travel_Frequently	Sales	2	Secundaria	Medical	Muy_Alta	Media	2
2068	34	0	Travel_Rarely	Research & Development	8	Universitaria	Medical	Media	Muy_Alta	2

1470 rows × 28 columns

In [47]:
```
df.to_excel('abandono_con_scoring.xlsx')
```