Fase 1: Entendimiento del Negocio (Business Understanding) 🎯

Bank Marketing Dataset - Data Dictionary

Dataset Information

Title: Bank Marketing (with social/economic context)

Source: UCI Machine Learning Repository **Created by:** Sérgio Moro (ISCTE-IUL), Paulo Cortez (Univ. Minho) and Paulo Rita (ISCTE-IUL) @ 2014

Citation

If you use this dataset, please include the following citation:

> [Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, In press, http://dx.doi.org/10.1016/j.dss.2014.03.001

Dataset Overview

- **Number of Instances:** 41,188 (bank-additional-full.csv)
- **Number of Features:** 20 input variables + 1 output variable
- **Time Period:** May 2008 to November 2010
- **Target Variable:** Binary classification - predict if the client will subscribe a bank term deposit

Feature Descriptions

Bank Client Data

Variable	Type	Description	Values
`age`	Numeric	Age of the client	Integer values
`job`	Categorical	Type of job	"admin.", "blue-collar", "entrepreneur", "housemaid", "management", "retired", "self-employed", "services", "student", "technician", "unemployed", "unknown"
`marital`	Categorical	Marital status	"divorced", "married", "single", "unknown" *Note: "divorced" includes divorced or widowed*
`education`	Categorical	Education level	"basic.4y", "basic.6y", "basic.9y", "high.school", "illiterate", "professional.course", "university.degree", "unknown"
`default`	Categorical	Has credit in default?	"no", "yes", "unknown"
`housing`	Categorical	Has housing loan?	"no", "yes", "unknown"
`loan`	Categorical	Has personal loan?	"no", "yes", "unknown"

Last Contact Information (Current Campaign)

Variable	Type	Description	Values
`contact`	Categorical	Contact communication type	"cellular", "telephone"

Variable	Type	Description	Values
`month`	Categorical	Last contact month of year	"jan", "feb", "mar", "apr", "may", "jun", "jul", "aug", "sep", "oct", "nov", "dec"
`day_of_week`	Categorical	Last contact day of the week	"mon", "tue", "wed", "thu", "fri"
`duration`	Numeric	Last contact duration (seconds)	Integer values ⚠️ **Important: See note below**

Campaign Information

Variable	Type	Description	Values
`campaign`	Numeric	Number of contacts performed during this campaign for this client	Integer (includes last contact)
`pdays`	Numeric	Days since client was last contacted from a previous campaign	Integer (999 = not previously contacted)
`previous`	Numeric	Number of contacts performed before this campaign for this client	Integer
`poutcome`	Categorical	Outcome of the previous marketing campaign	"failure", "nonexistent", "success"

Social and Economic Context Attributes

Variable	Type	Description	Indicator Type
`emp.var.rate`	Numeric	Employment variation rate	Quarterly indicator
`cons.price.idx`	Numeric	Consumer price index	Monthly indicator
`cons.conf.idx`	Numeric	Consumer confidence index	Monthly indicator
`euribor3m`	Numeric	Euribor 3 month rate	Daily indicator
`nr.employed`	Numeric	Number of employees	Quarterly indicator

Target Variable

Variable	Type	Description	Values
`y`	Binary	Has the client subscribed a term deposit?	"yes", "no"

Important Notes

Duration Variable Warning

⚠️ **Critical Note about `duration` variable:**

- This attribute highly affects the output target (if duration=0 then y="no")
- Duration is not known before a call is performed
- After the call ends, the outcome (y) is obviously known
- **Recommendation:** This variable should only be included for benchmark purposes and should be **discarded** if the intention is to have a realistic predictive model

Missing Values

- Several categorical attributes contain missing values

- All missing values are coded with the label **"unknown"**
- These can be treated as:
  - A possible class label
  - Handled using deletion techniques
  - Handled using imputation techniques

Dataset Versions

The dataset comes in two versions:

1. **bank-additional-full.csv**: Full dataset with 41,188 examples (ordered by date)
2. **bank-additional.csv**: 10% sample with 4,119 examples (randomly selected)
   - Provided for testing computationally demanding algorithms (e.g., SVM)

Additional Resources

- Original Paper
- UCI Repository
- Banco de Portugal Statistics

Data Dictionary 📊

Variable Name	Role	Type	Description
age	Feature	Numeric 🔢	Client's age.
job	Feature	Categorical 🏷️	Type of job: "admin.", "unknown", "unemployed", "management", "housemaid", "entrepreneur", "student", "blue-collar", "self-employed", "retired", "technician", "services".
marital	Feature	Categorical 🏷️	Marital status: "married", "divorced", "single". Note: "divorced" includes widowed. 💔
education	Feature	Categorical 🏷️	Education level: "unknown", "secondary", "primary", "tertiary". 🎓
default	Feature	Binary ✅❌	Has credit in default?: "yes" or "no". ⚠️
balance	Feature	Numeric 🔢	Average yearly balance. 💰
housing	Feature	Binary ✅❌	Has a housing loan?: "yes" or "no". 🏠
loan	Feature	Binary ✅❌	Has a personal loan?: "yes" or "no". 💳
contact	Feature	Categorical 🏷️	Contact communication type: "unknown", "telephone", "cellular". 📞📱
day	Feature	Numeric 🔢	Last contact day of the month. 🗓️
month	Feature	Categorical 🏷️	Last contact month of the year: "jan", "feb", "mar", ..., "nov", "dec". 🗓️
duration	Feature	Numeric 🔢	Last contact duration. ⏱️
campaign	Feature	Numeric 🔢	Number of contacts performed during this campaign for this client. 📢
pdays	Feature	Numeric 🔢	Days since client was last contacted from a previous campaign (-1 means not previously contacted). 🗓️
previous	Feature	Numeric 🔢	Number of contacts performed before this campaign for this client. 🔁
poutcome	Feature	Categorical 🏷️	Outcome of the previous marketing campaign: "unknown", "other", "failure", "success". 📈📉
y	Target	Binary ✅❌	Has the client subscribed to a term deposit?: "yes" or "no". 🎯

Legend 📝

- 🔢 **Numeric**: Continuous or discrete numerical values
- 🏷️ **Categorical**: Discrete categories or labels
- ✅ ❌ **Binary**: Yes/No or True/False values
- 🎯 **Target**: The variable we want to predict
- ⚠️ **Important**: Credit default status - critical risk indicator
- 💰 **Financial**: Money-related variable
- 📞 📱 **Communication**: Contact method indicators
- ⏱️ **Time**: Duration or time-related measurements

In [1]:
```
import pandas as pd
from tools import *
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
import statsmodels.api as sm
import statsmodels.formula.api as smf
from ydata_profiling import ProfileReport
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

# Set pandas display options for better readability
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
pd.set_option('display.max_colwidth', 50)



# Configuración de visualización
plt.style.use('default')
sns.set_palette("husl")
plt.rcParams['figure.figsize'] = (12, 8)
plt.rcParams['font.size'] = 12


plt.style.use('seaborn-v0_8')
```


In [2]:
```
print("Step 1: Loading and preprocessing data...")
df = load_and_preprocess_data('data/bank-additional-full.csv')
basic_info = get_basic_info(df)
```

```
Step 1: Loading and preprocessing data...
```

Actividad: Análisis Univariado de una Variable en un Conjunto de Datos

Descripción

En esta actividad, los estudiantes deberán seleccionar una columna (variable) de un conjunto de datos, describir su importancia, realizar un análisis univariado utilizando Python y presentar conclusiones basadas en los hallazgos.

Comparte el enlace a tu notebook en tu repositorio de github (puede ser el enlace a tu github con acceso público o sólo a armandoordonez@gmail.com)

Instrucciones

1. Descripción de la importancia de la columna (1 punto)

- Explicar por qué la variable seleccionada es relevante dentro del conjunto de datos.
- Indicar cómo podría influir en el análisis o en la toma de decisiones.

2. Análisis univariado en Python (2 puntos)

- Generar estadísticas descriptivas (media, mediana, moda, desviación estándar, valores atípicos, etc.).
- Visualizar la distribución de la variable usando histogramas, boxplots u otras gráficas adecuadas.
- Identificar posibles sesgos o patrones dentro de los datos.

3. Conclusiones del análisis (2 puntos)

- Resumir los hallazgos clave obtenidos en el análisis.
- Mencionar implicaciones o próximos pasos que podrían derivarse de los resultados.

Ejemplo de estructura de entrega:

1. **Introducción y selección de la variable**
2. **Explicación de su importancia**
3. **Código en Python con análisis univariado y visualizaciones**
4. **Interpretación de los resultados**
5. **Conclusiones**

Criterios de Calificación

Criterio	Puntos
Describe claramente la importancia de la columna seleccionada para el problema de ciencia de datos o IA	1 pts
Realiza el análisis estadístico y gráfico univariado en Python	2 pts
Las conclusiones están alineadas con el análisis univariado realizado	2 pts
Total	**5 pts**

Taller 2 Solution

In [3]:
```
print("Análisis Descriptivo de la Columna 'marital':")
print("-" * 45)

# 1. Frecuencia Absoluta (Absolute Frequency)
# Contamos cuántas veces aparece cada categoría en la columna.
print("\n1. Frecuencia Absoluta (Conteo de Valores):")
marital_counts = df['marital'].value_counts()
print(marital_counts)
```

```
Análisis Descriptivo de la Columna 'marital':
--------------------------------------------

1. Frecuencia Absoluta (Conteo de Valores):
marital
married     24928
single      11568
divorced     4612
unknown        80
Name: count, dtype: int64
```

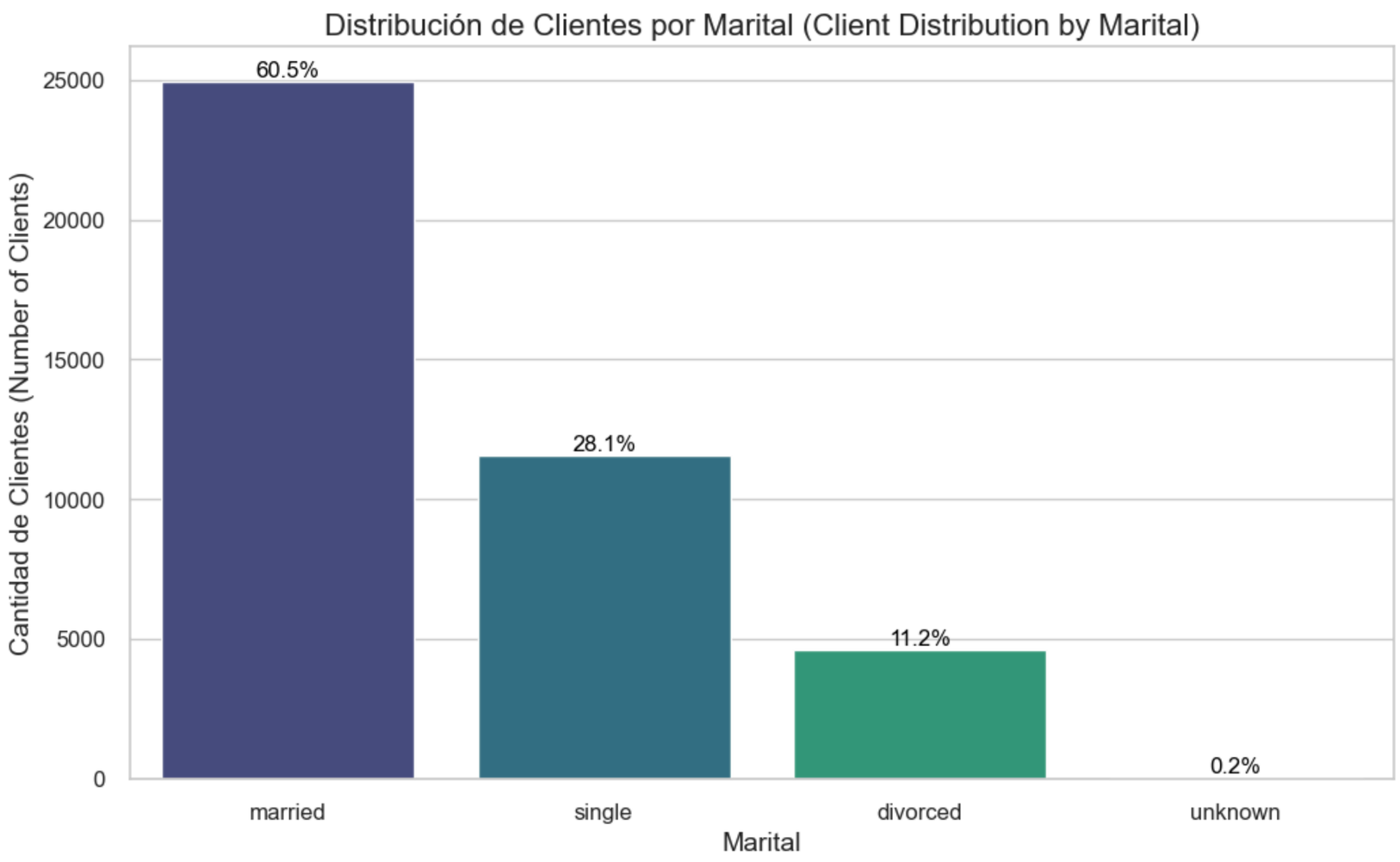
In [4]:
```python
print("\n2. Frecuencia Relativa (Porcentaje):")
marital_percentages = df['marital'].value_counts(normalize=True) * 100
# Usamos .round(2) para redondear a dos decimales y lo mostramos como texto con el símbolo '%'.
print(marital_percentages.round(2).astype(str) + ' %')
```

```
2. Frecuencia Relativa (Porcentaje):
marital
married     60.52 %
single      28.09 %
divorced     11.2 %
unknown      0.19 %
Name: proportion, dtype: object
```

In [5]:
```python
print("\n3. Resumen Rápido con describe():")
marital_summary = df['marital'].describe()
print(marital_summary)
```

```
3. Resumen Rápido con describe():
count       41188
unique          4
top       married
freq        24928
Name: marital, dtype: object
```

In [6]:
```python
# --- Visualización (Visualization) ---
# Llamamos a la función encapsulada para generar el gráfico.
plot_distribution(df, 'marital')
```



Taller 2. Soluccion (Comentarios / Análisis Univariado de una Variable en un Conjunto de Datos)

Para el análisis univariado de este proyecto, me enfoqué en explorar la variable **marital** (estado civil). El análisis revela una clara predominancia de ciertos grupos de clientes, distribuidos de la siguiente manera:

Distribución por Estado Civil

- **Married (Casados)**: 60.5%
- **Single (Solteros)**: 28.1%
- **Divorced (Divorciados)**: 11.2%
- **Unknown (Desconocido)**: 0.2%

> **Nota**: La categoría *unknown* es estadísticamente insignificante (0.2%), por lo que sus datos podrían gestionarse más adelante con alguna de las técnicas de imputación vistas en el curso.

Interpretación Preliminar

Estos datos sugieren que las **campañas de marketing** deberían enfocarse prioritariamente en el perfil demográfico de las personas casadas, ya que representan la base de clientes más sustancial.

Paralelamente, sería valioso comprender la psicología y el comportamiento de este grupo para que, mediante la **ingeniería de características**, se puedan desarrollar variables más precisas y predictivas.

Actividad: Análisis Bivariado en Python

Objetivo:

Los estudiantes aplicarán técnicas de análisis exploratorio de datos (EDA) para examinar la relación entre dos variables en un conjunto de datos, utilizando Python. Deberán interpretar los resultados y extraer conclusiones relevantes.

Comparte el enlace a tu notebook en github.

Instrucciones:

Realización del Análisis Bivariado en Python (3 puntos)

1. **Seleccionar dos variables** de un conjunto de datos y **justificar su elección**.
2. **Calcular estadísticas** que describan la relación entre las variables (correlación, tablas de contingencia, etc.).
3. **Generar visualizaciones apropiadas**, como:
   - Diagramas de dispersión (para variables numéricas).
   - Boxplots comparativos (para una variable numérica y una categórica).
   - Heatmaps de correlación (para múltiples variables numéricas).
   - Gráficos de barras o stacked charts (para variables categóricas).
4. **Interpretar las tendencias y patrones observados**.

Conclusiones (2 puntos)

- Resumir los hallazgos clave del análisis.
- Explicar el impacto de la relación entre las variables en el contexto del conjunto de datos.
- Plantear posibles hipótesis o próximos pasos para un análisis más profundo.

Fase 2:Entendimiento de los Datos (Data Understanding) 🔍

In [7]:
```
# Step 2: Data quality analysis
print("Step 2: Analyzing data quality...")
quality_analysis = analyze_data_quality(df)
var_types = identify_variable_types(df)
```

```
Step 2: Analyzing data quality...
```

In [8]:
```
# Step 3: Target variable analysis
print("Step 3: Analyzing target variable...")
target_analysis = analyze_target_variable(df)
```

```
Step 3: Analyzing target variable...
```

In [9]:
```
# Step 4: Numeric variables analysis
print("Step 4: Analyzing numeric variables...")
numeric_analysis_general = analyze_numeric_variables(df, var_types['numeric'])
```

```
Step 4: Analyzing numeric variables...
```

In [10]:
```
numeric_analysis_general
```

Out[10]: {'stats':   age   duration   campaign   pdays   previous \
count 41188.00000 41188.000000 41188.000000 41188.000000 41188.000000
mean   40.02406   258.285010   2.567593   962.475454   0.172963
std   10.42125   259.279249   2.770014   186.910907   0.494901
min   17.00000   0.000000   1.000000   0.000000   0.000000
25%   32.00000   102.000000   1.000000   999.000000   0.000000
50%   38.00000   180.000000   2.000000   999.000000   0.000000
75%   47.00000   319.000000   3.000000   999.000000   0.000000
max   98.00000   4918.000000   56.000000   999.000000   7.000000

       emp.var.rate cons.price.idx cons.conf.idx   euribor3m   nr.employed
count 41188.000000   41188.000000   41188.000000 41188.000000 41188.000000
mean   0.081886   93.575664   -40.502600   3.621291   5167.035911
std   1.570960   0.578840   4.628198   1.734447   72.251528
min   -3.400000   92.201000   -50.800000   0.634000   4963.600000
25%   -1.800000   93.075000   -42.700000   1.344000   5099.100000
50%   1.100000   93.749000   -41.800000   4.857000   5191.000000
75%   1.400000   93.994000   -36.400000   4.961000   5228.100000
max   1.400000   94.767000   -26.900000   5.045000   5228.100000 ,
'variable_vars': ['age',
 'duration',
 'campaign',
 'pdays',
 'previous',
 'emp.var.rate',
 'cons.price.idx',
 'cons.conf.idx',
 'euribor3m',
 'nr.employed'],
'distribution_info': {'age': {'skewness': np.float64(0.7846682380932289),
  'kurtosis': np.float64(0.7910698035273853),
  'normality_test': np.float64(0.0)},
 'duration': {'skewness': np.float64(3.2630224157610432),
  'kurtosis': np.float64(20.245334438614844),
  'normality_test': np.float64(0.0)},
 'campaign': {'skewness': np.float64(4.762333252560971),
  'kurtosis': np.float64(36.97516047878921),
  'normality_test': np.float64(0.0)},
 'pdays': {'skewness': np.float64(-4.922010656450045),
  'kurtosis': np.float64(22.2266185118177),
  'normality_test': np.float64(0.0)},
 'previous': {'skewness': np.float64(3.8319026847007014),
  'kurtosis': np.float64(20.106229526902816),
  'normality_test': np.float64(0.0)},
 'emp.var.rate': {'skewness': np.float64(-0.7240691785977529),
  'kurtosis': np.float64(-1.0626482012872505),
  'normality_test': np.float64(0.0)},
 'cons.price.idx': {'skewness': np.float64(-0.23087924271991117),
  'kurtosis': np.float64(-0.8298535163032219),
  'normality_test': np.float64(0.0)},
 'cons.conf.idx': {'skewness': np.float64(0.3031688173189229),
  'kurtosis': np.float64(-0.35866045494457843),
  'normality_test': np.float64(4.6668213776494045e-204)},
 'euribor3m': {'skewness': np.float64(-0.7091621286441162),
  'kurtosis': np.float64(-1.4067775199378407),
  'normality_test': np.float64(0.0)},
 'nr.employed': {'skewness': np.float64(-1.0442243763652297),
  'kurtosis': np.float64(-0.003905589182342073),
  'normality_test': np.float64(0.0)}}}

In [11]:
```
# Step 5: Categorical variables analysis
print("Step 5: Analyzing categorical variables...")
categorical_analysis_general = analyze_categorical_variables(df, var_types['categorical'])
```

```
Step 5: Analyzing categorical variables...
```

In [12]:
```
categorical_analysis_general
```

Out[12]:

```
{'job': {'value_counts': {'admin.': 10422,
   'blue-collar': 9254,
   'technician': 6743,
   'services': 3969,
   'management': 2924,
   'retired': 1720,
   'entrepreneur': 1456,
   'self-employed': 1421,
   'housemaid': 1060,
   'unemployed': 1014,
   'student': 875,
   'unknown': 330},
  'proportions': {'admin.': 25.303486452364766,
   'blue-collar': 22.46770904146839,
   'technician': 16.37127318636496,
   'services': 9.636301835486064,
   'management': 7.099155093716616,
   'retired': 4.175973584539186,
   'entrepreneur': 3.5350101971447994,
   'self-employed': 3.450033990482665,
   'housemaid': 2.5735651160532194,
   'unemployed': 2.4618821015829853,
   'student': 2.1244051665533648,
   'unknown': 0.8012042342429834},
  'conversion_rates': {'admin.': 12.97,
   'blue-collar': 6.89,
   'entrepreneur': 8.52,
   'housemaid': 10.0,
   'management': 11.22,
   'retired': 25.23,
   'self-employed': 10.49,
   'services': 8.14,
   'student': 31.43,
   'technician': 10.83,
   'unemployed': 14.2,
   'unknown': 11.21},
  'unique_count': 12},
 'marital': {'value_counts': {'married': 24928,
   'single': 11568,
   'divorced': 4612,
   'unknown': 80},
  'proportions': {'married': 60.52248227639119,
   'single': 28.085850247644945,
   'divorced': 11.197436146450421,
   'unknown': 0.1942313295134505},
  'conversion_rates': {'divorced': 10.32,
   'married': 10.16,
   'single': 14.0,
   'unknown': 15.0},
  'unique_count': 4},
 'education': {'value_counts': {'university.degree': 12168,
   'high.school': 9515,
   'basic.9y': 6045,
   'professional.course': 5243,
   'basic.4y': 4176,
   'basic.6y': 2292,
   'unknown': 1731,
   'illiterate': 18},
  'proportions': {'university.degree': 29.542585218995825,
   'high.school': 23.10138875400602,
   'basic.9y': 14.676604836360104,
```

```
 'professional.course': 12.729435757987764,
 'basic.4y': 10.138875400602117,
 'basic.6y': 5.5647275905603575,
 'unknown': 4.202680392347285,
 'illiterate': 0.04370204914052637},
'conversion_rates': {'basic.4y': 10.25,
 'basic.6y': 8.2,
 'basic.9y': 7.82,
 'high.school': 10.84,
 'illiterate': 22.22,
 'professional.course': 11.35,
 'university.degree': 13.72,
 'unknown': 14.5},
'unique_count': 8},
'default': {'value_counts': {'no': 32588, 'unknown': 8597, 'yes': 3},
 'proportions': {'no': 79.12013207730408,
 'unknown': 20.87258424783918,
 'yes': 0.007283674856754395},
 'conversion_rates': {'no': 12.88, 'unknown': 5.15, 'yes': 0.0},
 'unique_count': 3},
'housing': {'value_counts': {'yes': 21576, 'no': 18622, 'unknown': 990},
 'proportions': {'yes': 52.384189569777604,
 'no': 45.21219772749345,
 'unknown': 2.40361270272895},
 'conversion_rates': {'no': 10.88, 'unknown': 10.81, 'yes': 11.62},
 'unique_count': 3},
'loan': {'value_counts': {'no': 33950, 'yes': 6248, 'unknown': 990},
 'proportions': {'no': 82.42692046227057,
 'yes': 15.169466835000486,
 'unknown': 2.40361270272895},
 'conversion_rates': {'no': 11.34, 'unknown': 10.81, 'yes': 10.93},
 'unique_count': 3},
'contact': {'value_counts': {'cellular': 26144, 'telephone': 15044},
 'proportions': {'cellular': 63.47479848499563,
 'telephone': 36.52520151500437},
 'conversion_rates': {'cellular': 14.74, 'telephone': 5.23},
 'unique_count': 2},
'month': {'value_counts': {'may': 13769,
 'jul': 7174,
 'aug': 6178,
 'jun': 5318,
 'nov': 4101,
 'apr': 2632,
 'oct': 718,
 'sep': 570,
 'mar': 546,
 'dec': 182},
 'proportions': {'may': 33.429639700883754,
 'jul': 17.417694474118676,
 'aug': 14.999514421676215,
 'jun': 12.911527629406624,
 'nov': 9.956783529183259,
 'apr': 6.390210740992522,
 'oct': 1.7432261823832185,
 'sep': 1.383898222783335,
 'mar': 1.3256288239293,
 'dec': 0.4418762746430999},
 'conversion_rates': {'apr': 20.48,
 'aug': 10.6,
 'dec': 48.9,
 'jul': 9.05,
 'jun': 10.51,
```

```
  'mar': 50.55,
  'may': 6.43,
  'nov': 10.14,
  'oct': 43.87,
  'sep': 44.91},
 'unique_count': 10},
'day_of_week': {'value_counts': {'thu': 8623,
  'mon': 8514,
  'wed': 8134,
  'tue': 8090,
  'fri': 7827},
 'proportions': {'thu': 20.935709429931048,
  'mon': 20.67106924346897,
  'wed': 19.748470428280083,
  'tue': 19.641643197047685,
  'fri': 19.003107701272214},
 'conversion_rates': {'fri': 10.81,
  'mon': 9.95,
  'thu': 12.12,
  'tue': 11.78,
  'wed': 11.67},
 'unique_count': 5},
'poutcome': {'value_counts': {'nonexistent': 35563,
  'failure': 4252,
  'success': 1373},
 'proportions': {'nonexistent': 86.3431096435855,
  'failure': 10.323395163639896,
  'success': 3.3334951927745946},
 'conversion_rates': {'failure': 14.23,
  'nonexistent': 8.83,
  'success': 65.11},
 'unique_count': 3}}
```


In [13]:
```
selected_numeric='duration'
selected_categorical='job'
```

In [14]:
```
# Step 6: Focused bivariate analysis
print("Step 6: Performing bivariate analysis...")
numeric_bivariate = analyze_numeric_vs_target(df, selected_numeric)
categorical_bivariate = analyze_categorical_vs_target(df, selected_categorical)
```

```
Step 6: Performing bivariate analysis...
```

In [15]:
```
# Step 7: Correlation analysis
print("Step 7: Computing correlations...")
correlation_analysis = calculate_correlations(df, var_types['numeric'])
```

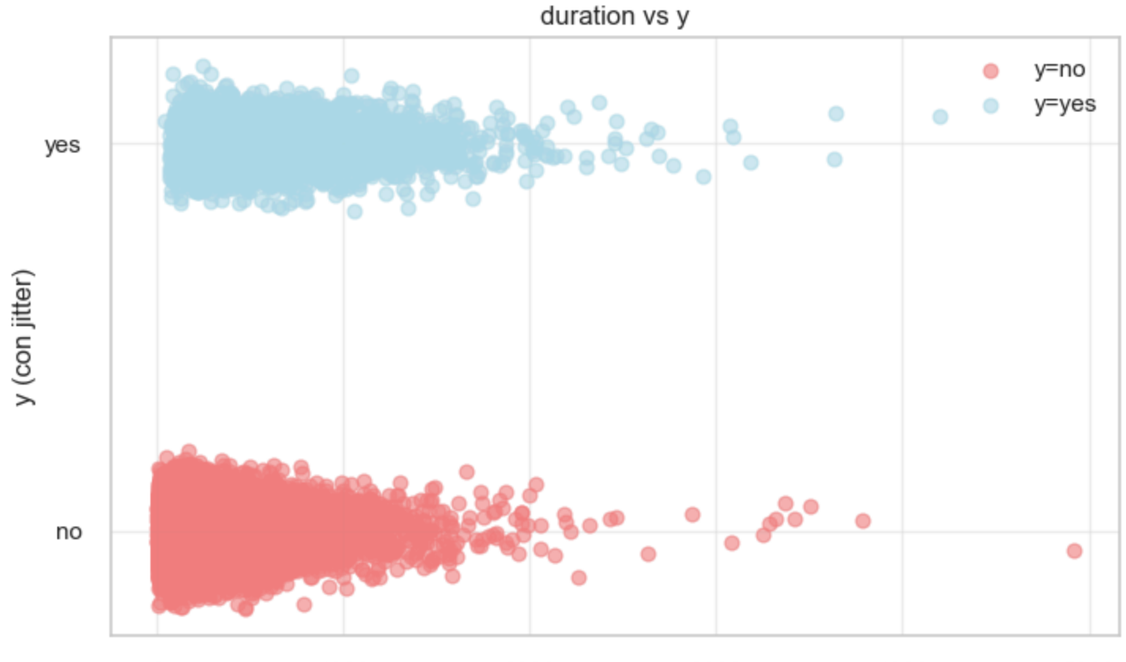
```
Step 7: Computing correlations...
```

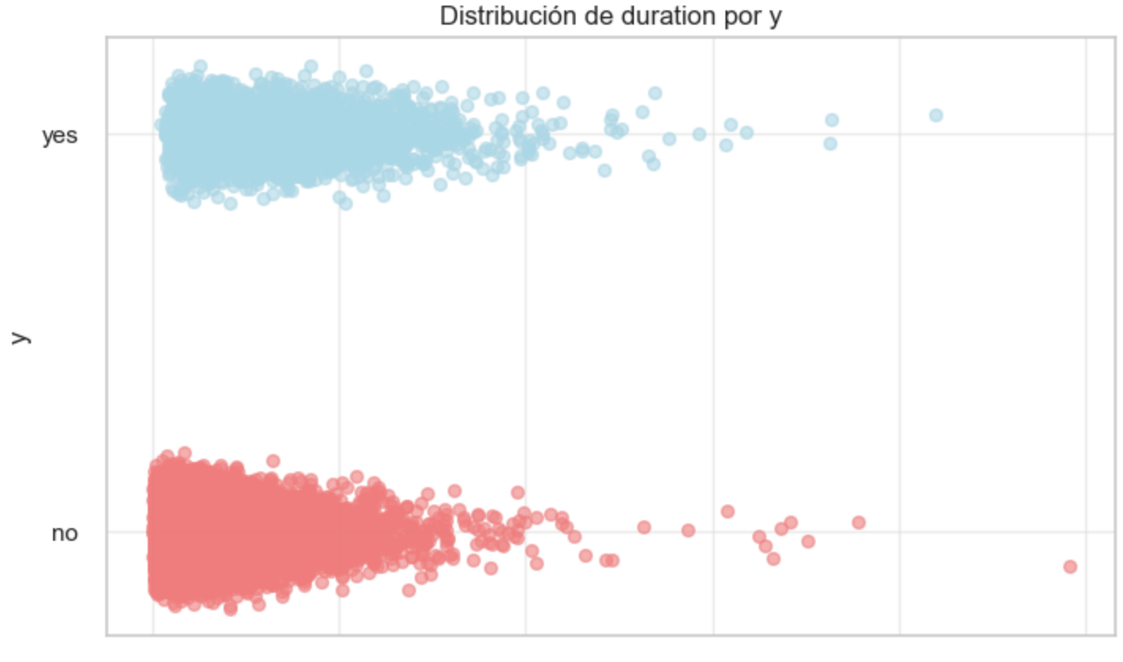
In [16]:
```
# Step 8: Generate visualizations
print("Step 8: Creating all required visualization types...")

print("  → Generando Diagramas de Dispersión (variables numéricas)...")
fig1 = create_scatter_plots_numeric(df, selected_numeric)

print("  → Generando Boxplots Comparativos (numérica vs categórica)...")

fig2 = create_comparative_boxplots(df, selected_numeric, selected_categorical)

print("  → Generando Heatmaps de Correlación (múltiples variables numéricas)...")
fig3 = create_correlation_heatmap(correlation_analysis)
```

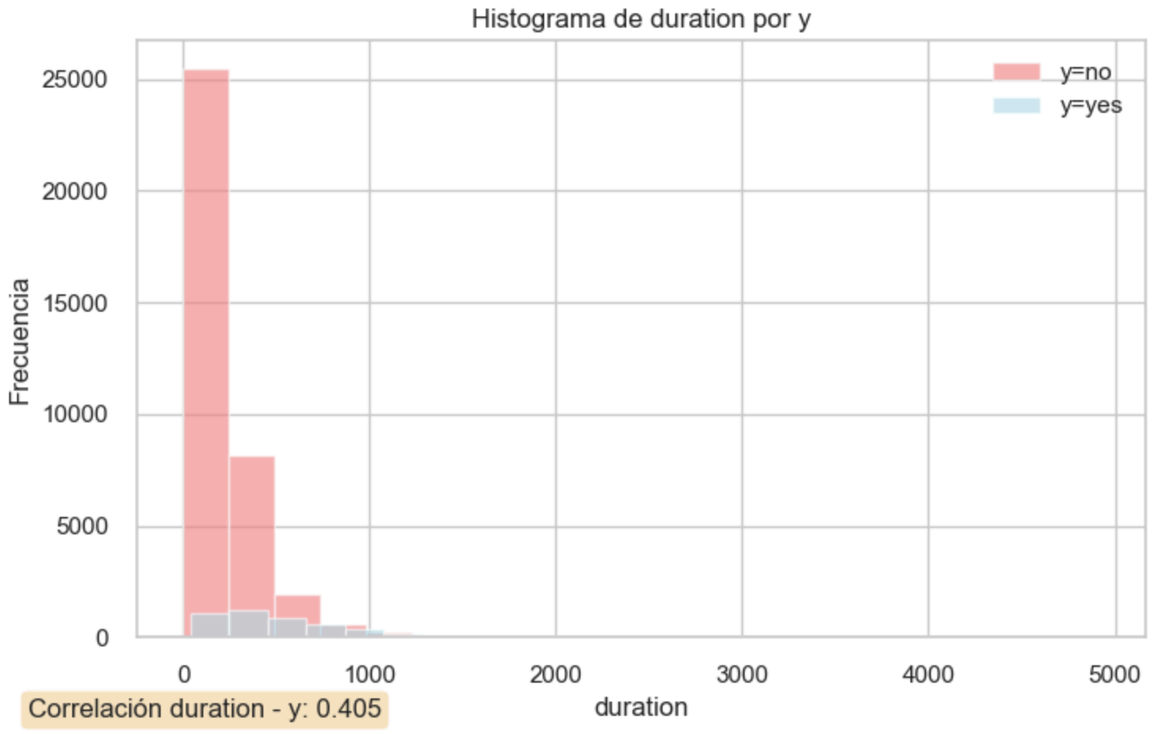
```
print("  → Generando Gráficos de Barras y Stacked Charts (variables categóricas)...")
fig4 = create_categorical_bar_charts(df, selected_categorical)

# Additional: Target distribution for context
print("  → Generando distribución de variable objetivo...")
fig5 = create_target_distribution_plot(target_analysis)

plt.show()
```

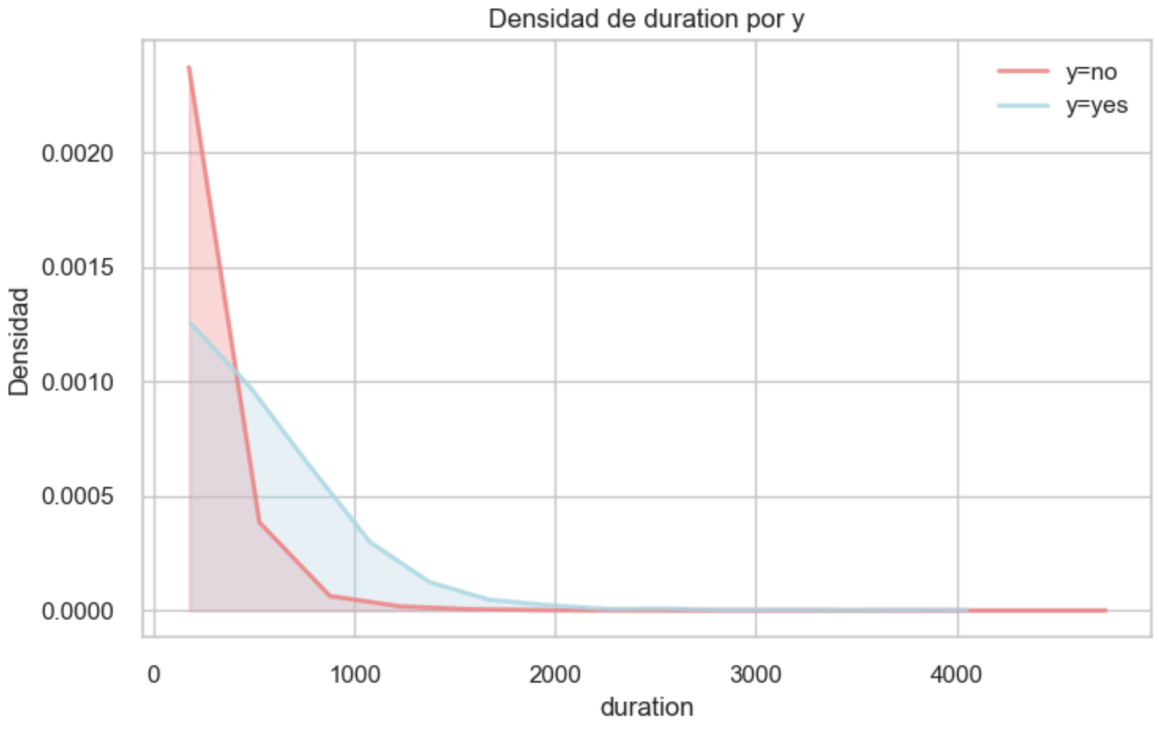
```
Step 8: Creating all required visualization types...
  → Generando Diagramas de Dispersión (variables numéricas)...
  → Generando Boxplots Comparativos (numérica vs categórica)...
  → Generando Heatmaps de Correlación (múltiples variables numéricas)...
  → Generando Gráficos de Barras y Stacked Charts (variables categóricas)...
  → Generando distribución de variable objetivo...
```

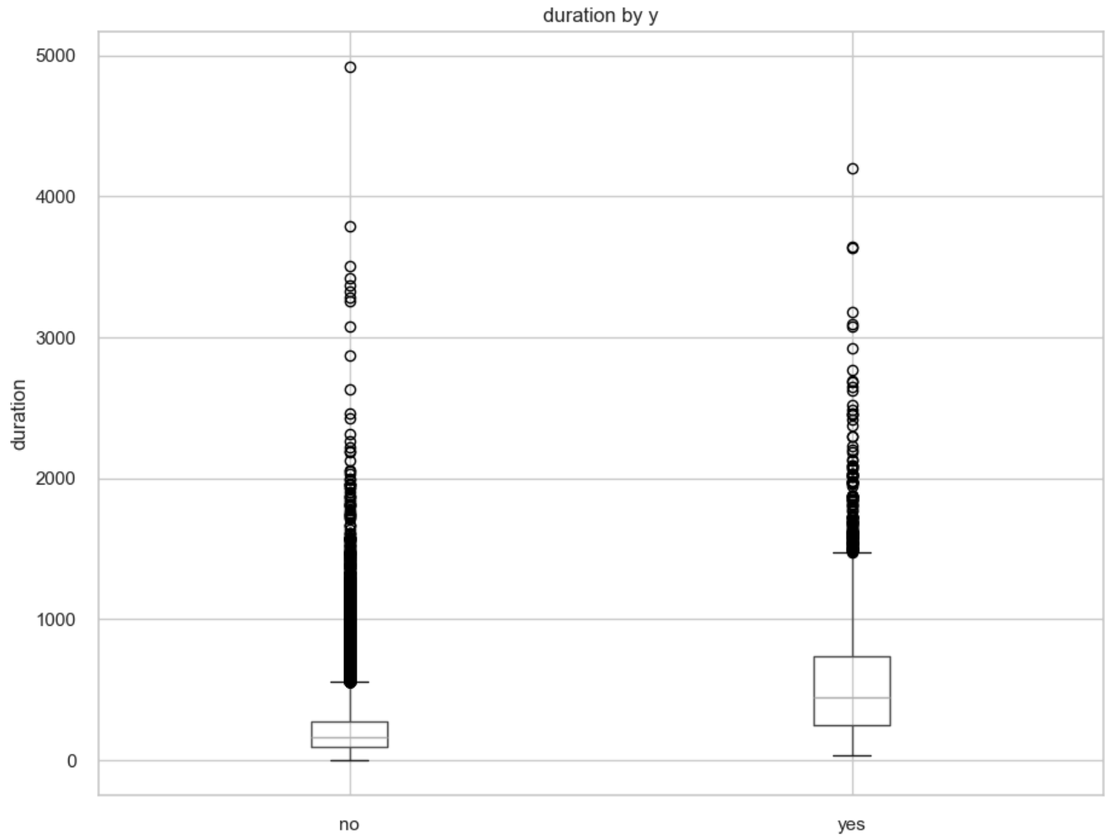
DIAGRAMAS DE DISPERSIÓN (Variable Numérica vs Target)

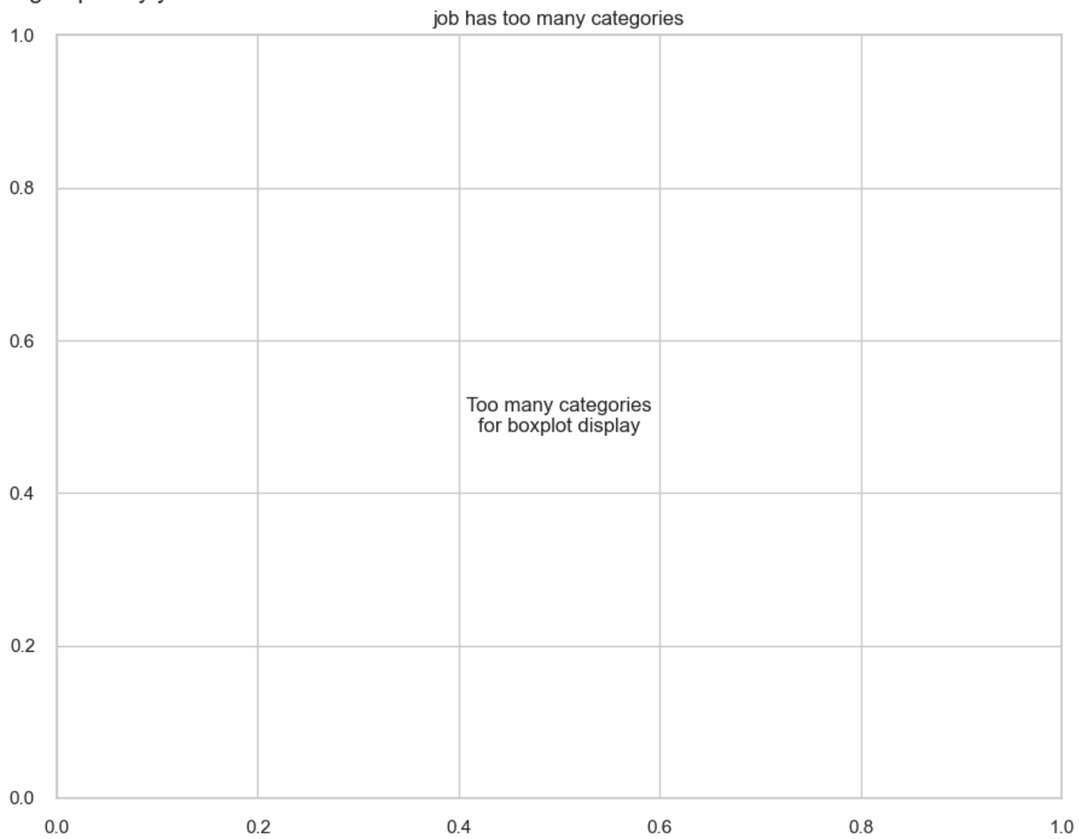


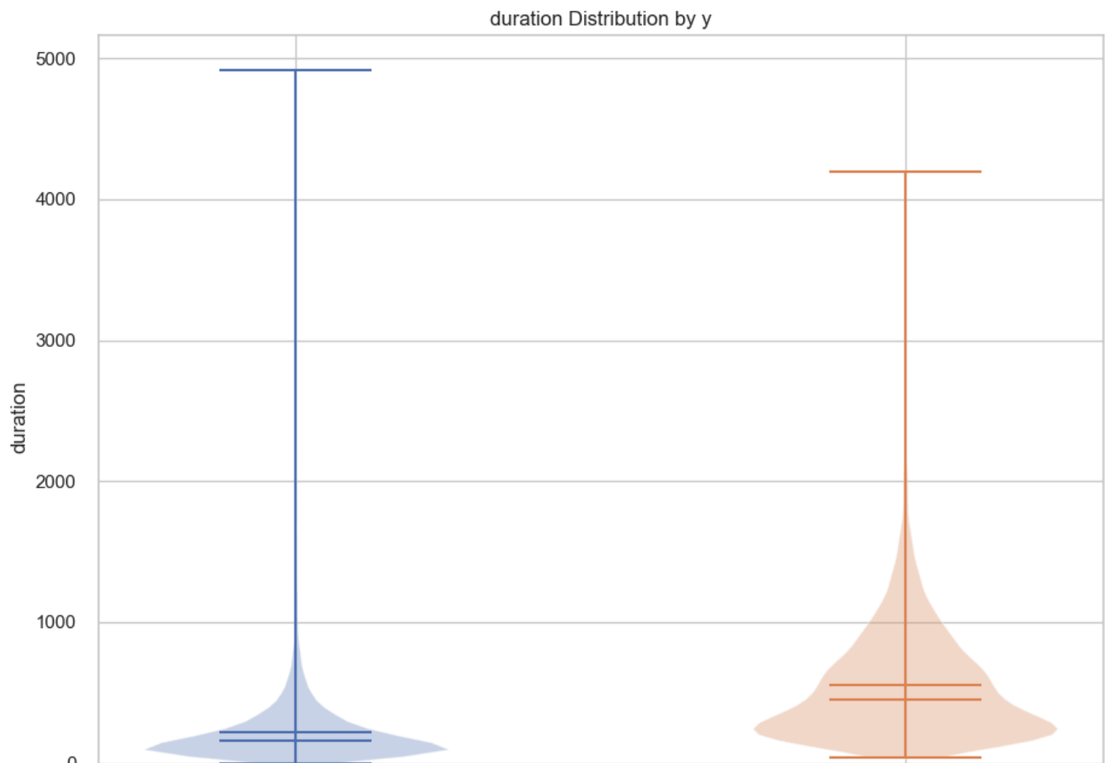


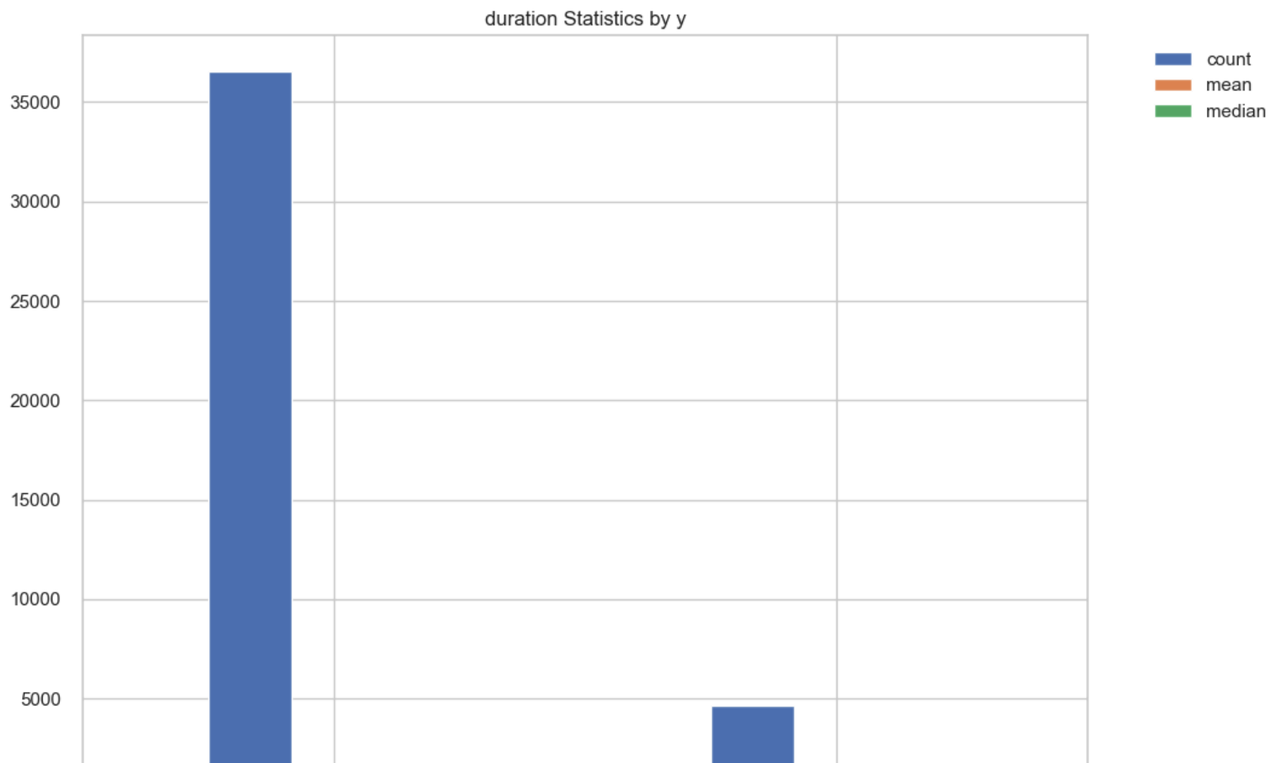


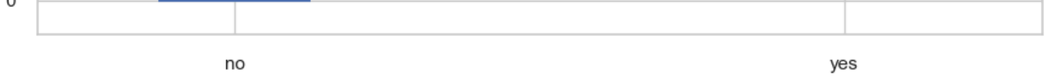



duration by y
duration
5000
4000
3000
2000
1000
0
no
yes
y

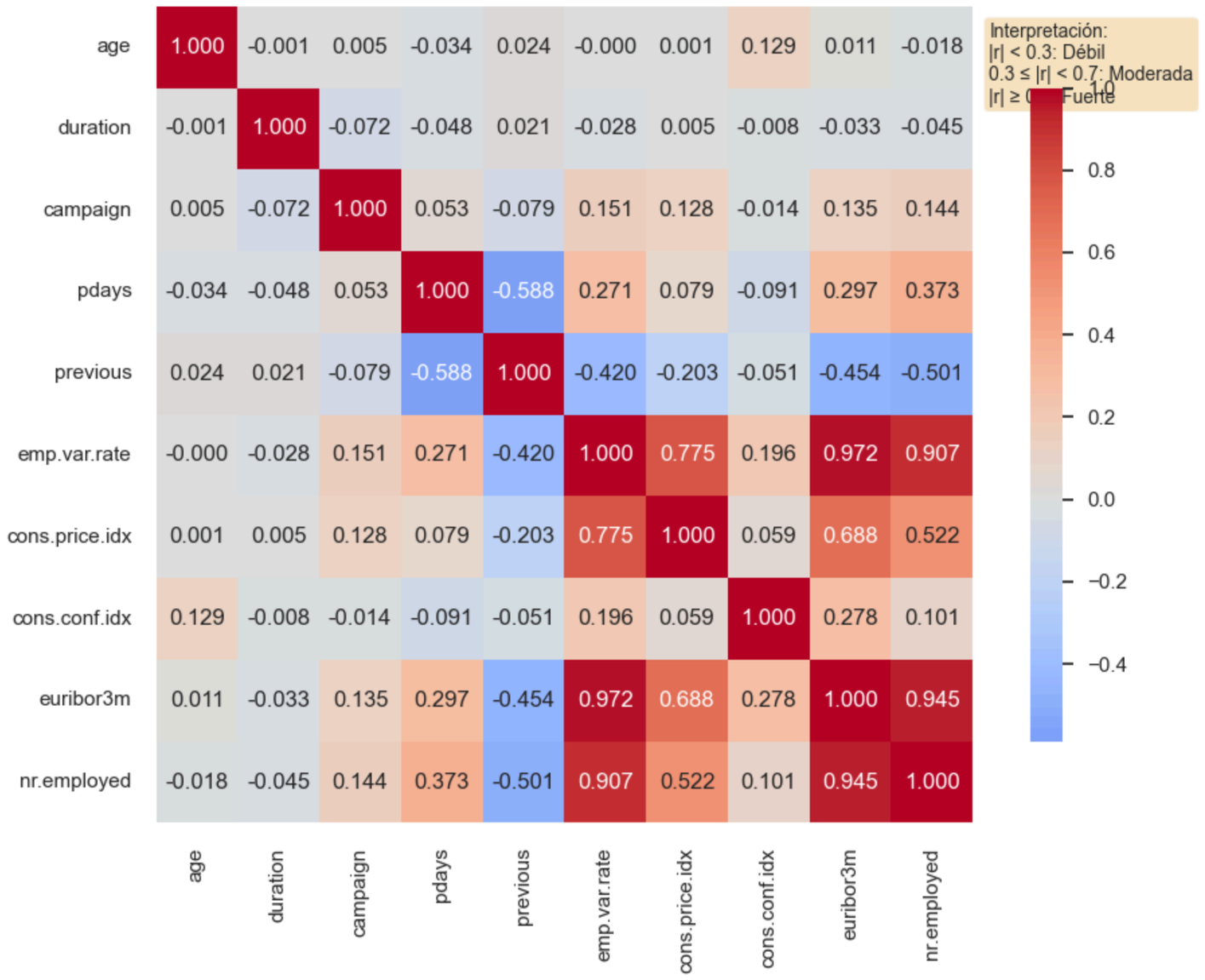

job has too many categories
Too many categories
for boxplot display
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


duration Distribution by y
duration
5000
4000
3000
2000
1000
0
no
yes
y

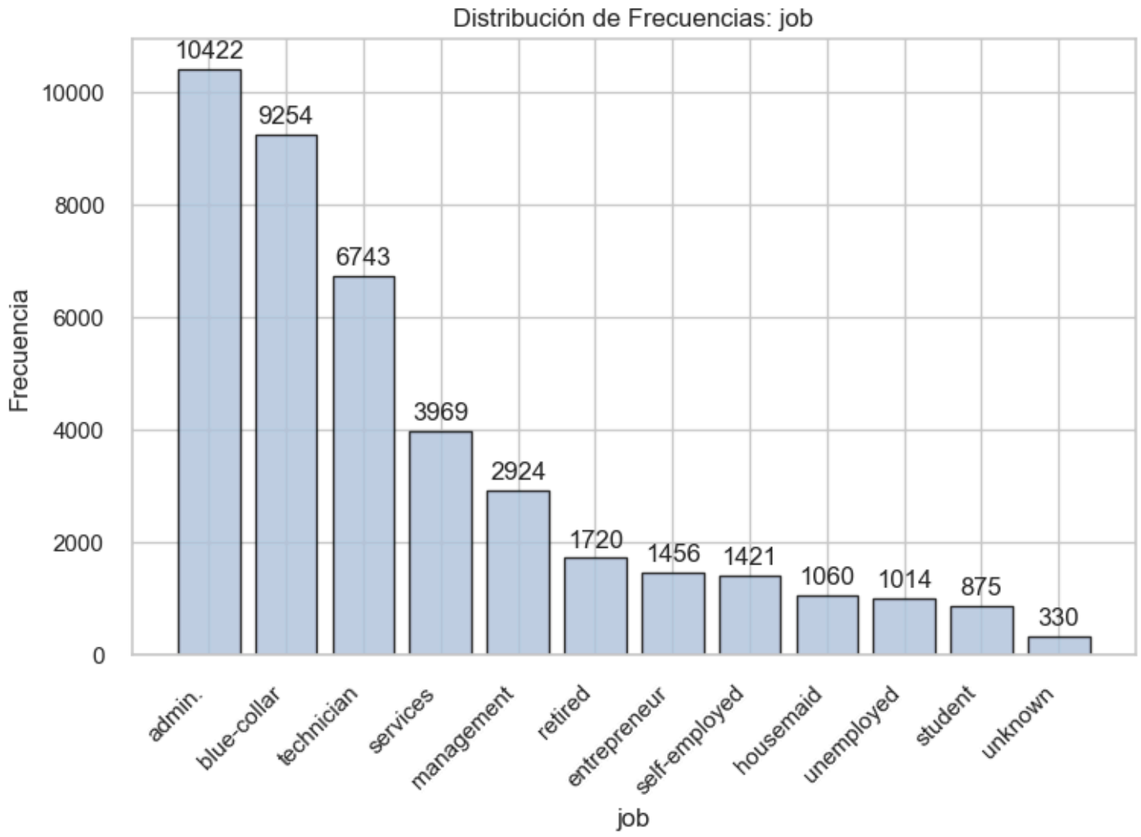

duration Statistics by y
count
mean
median
35000
30000
25000
20000
15000
10000
5000

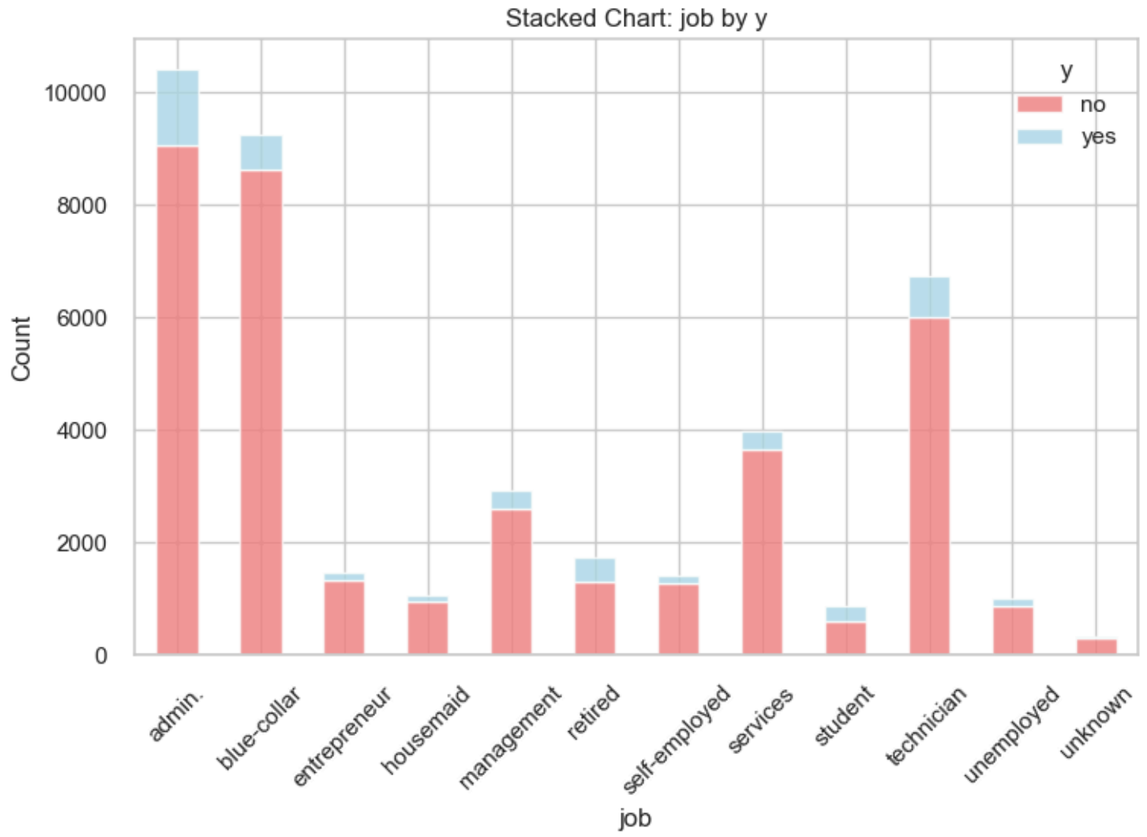


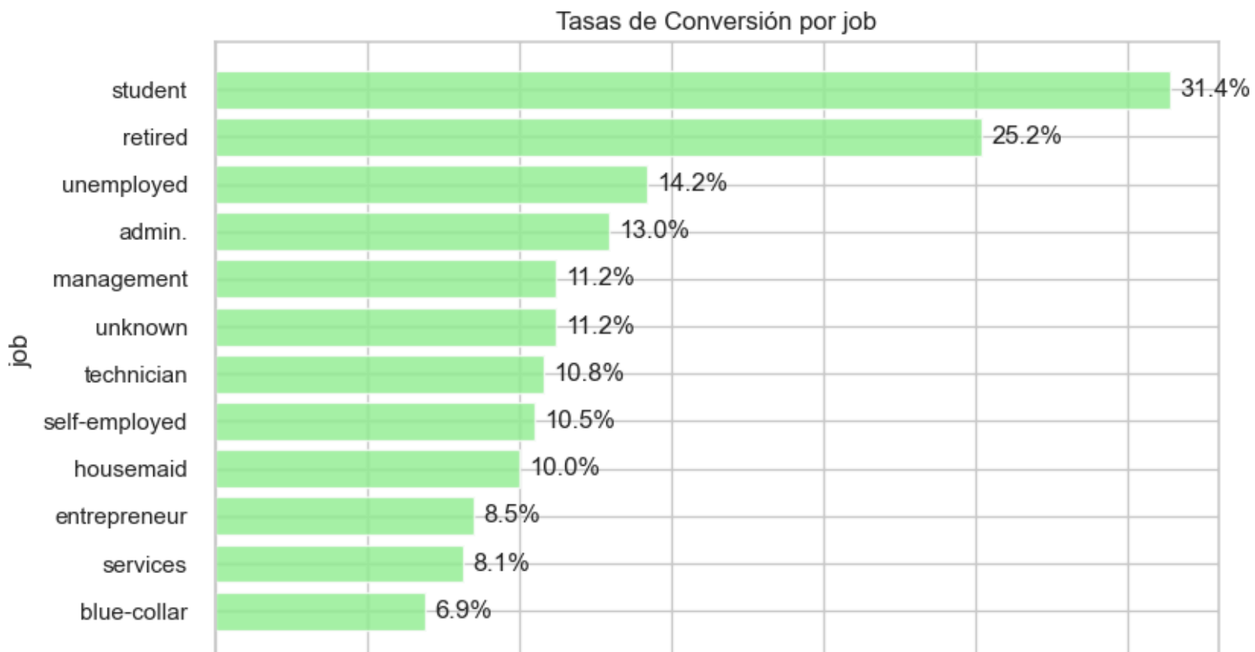
HEATMAP DE CORRELACIÓN (Variables Numéricas)

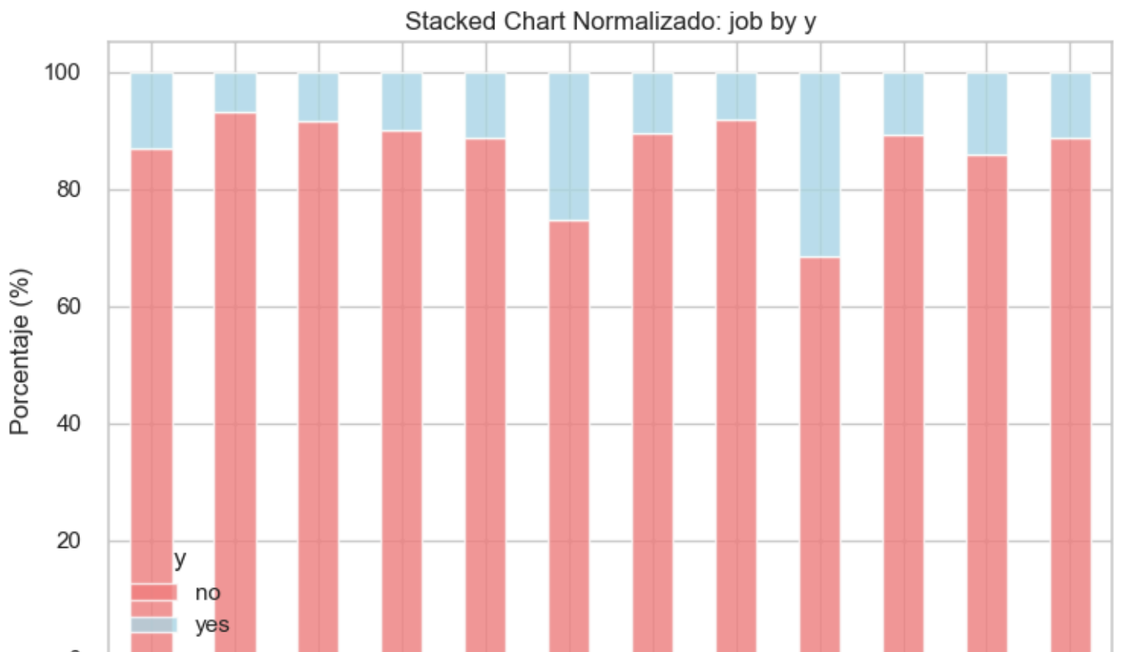


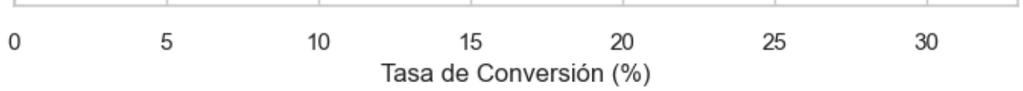
GRÁFICOS DE BARRAS Y STACKED CHARTS (Variables Categóricas)



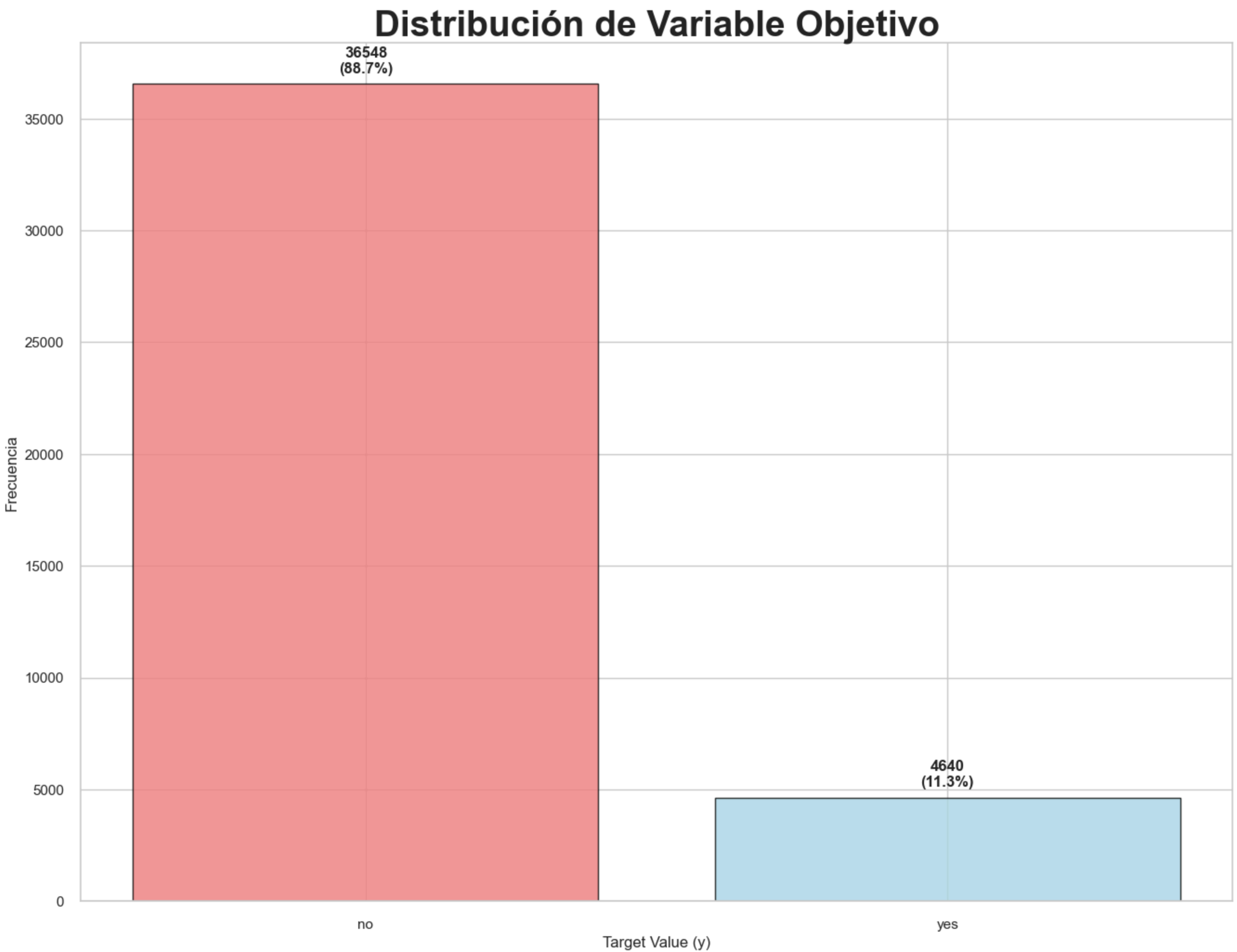






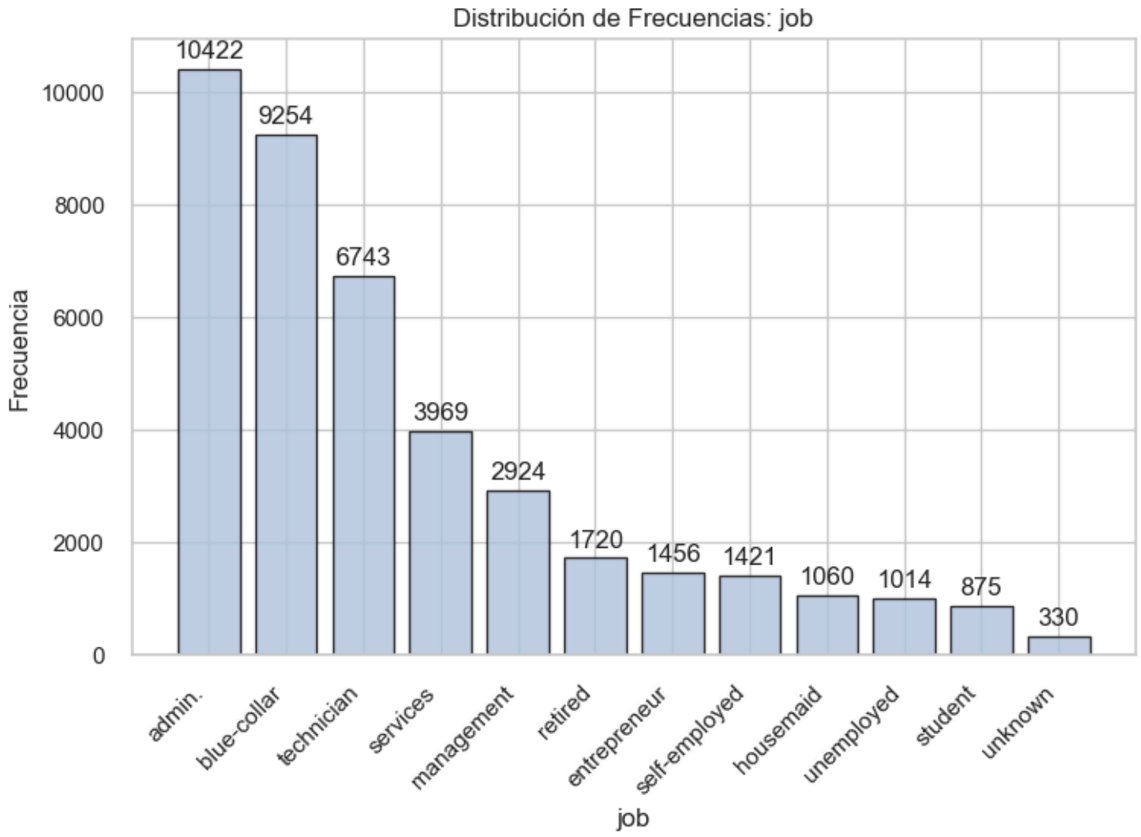

0
5
10
15
20
25
30
Tasa de Conversión (%)

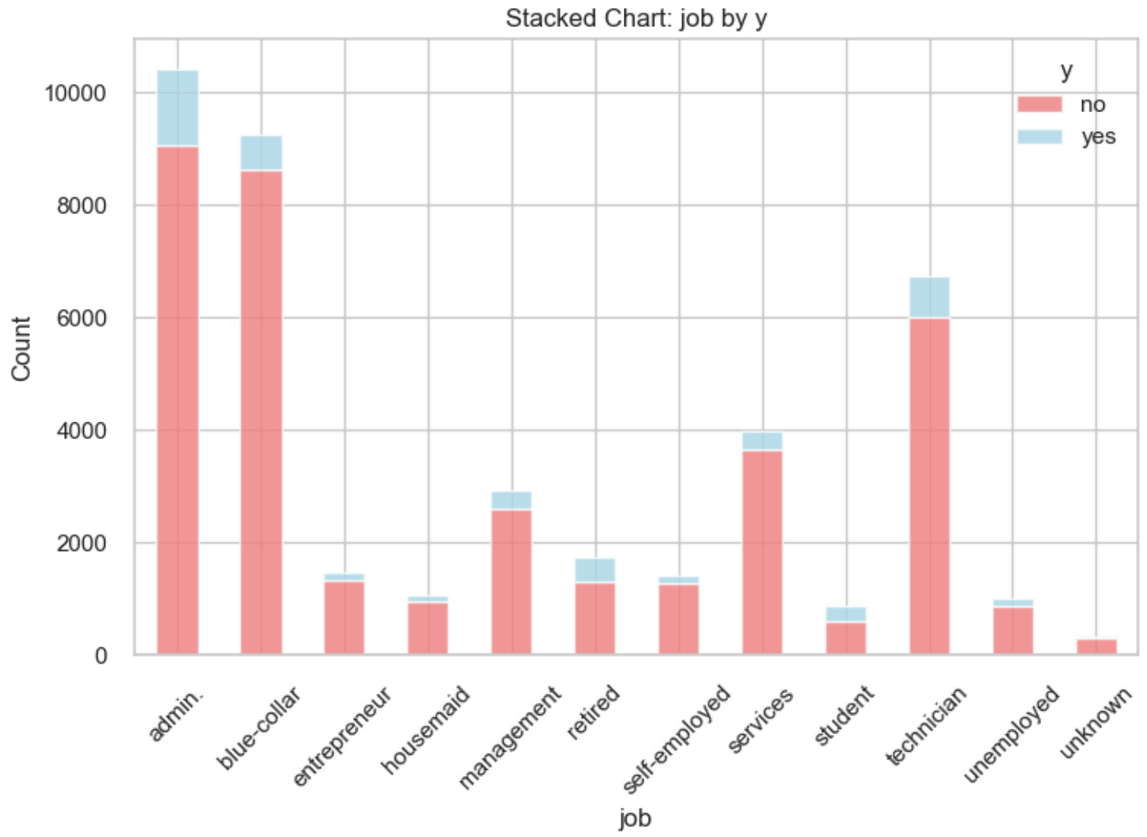

0
admin.
blue-collar
entrepreneur
housemaid
management
retired
self-employed
services
student
technician
unemployed
unknown
job

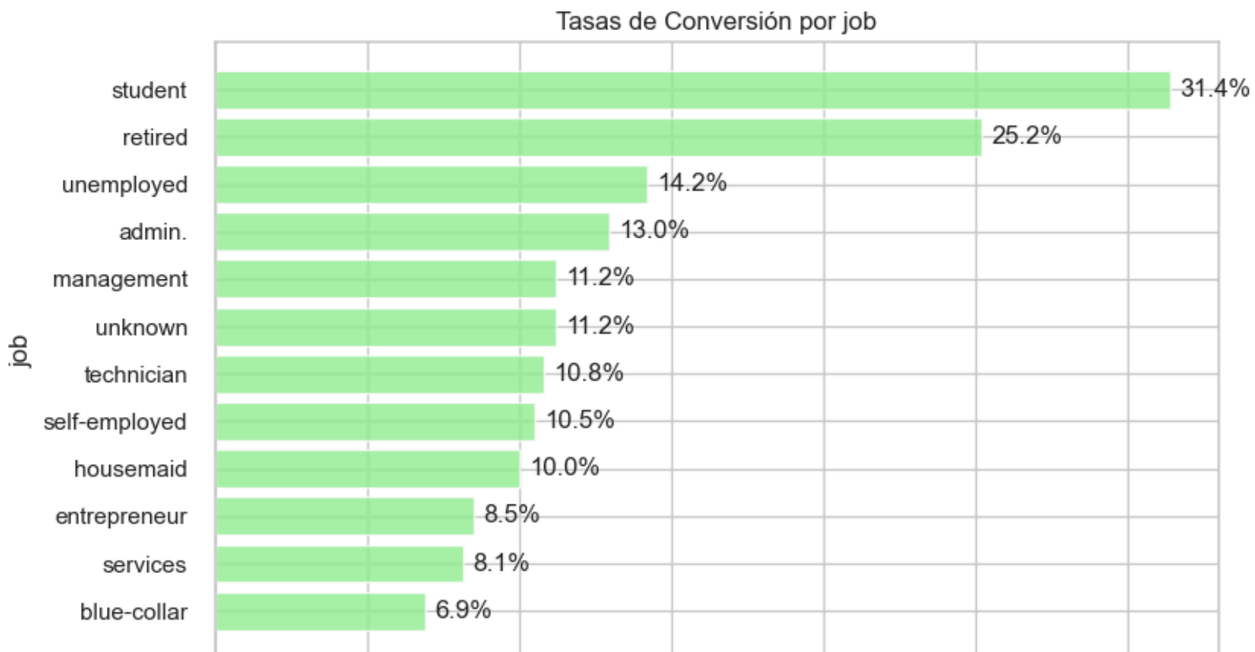

Distribución de Variable Objetivo
36548
(88.7%)
4640
(11.3%)
Frecuencia
0
5000
10000
15000
20000
25000
30000
35000
no
yes
Target Value (y)

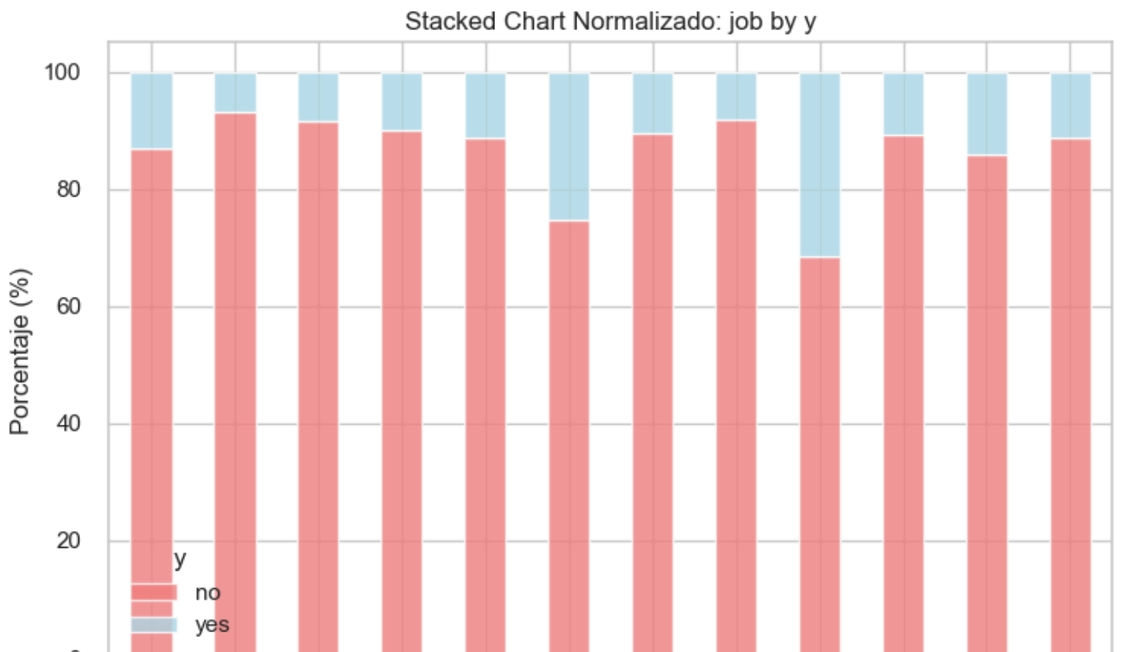
In [17]:
```
#### Analisis General todas las variables categoricas.
for current_category in categorical_analysis_general:
    fig4 = create_categorical_bar_charts(df, current_category)
plt.show()
```

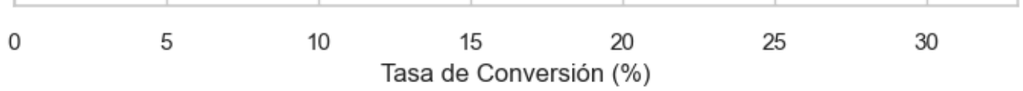
GRÁFICOS DE BARRAS Y STACKED CHARTS (Variables Categóricas)





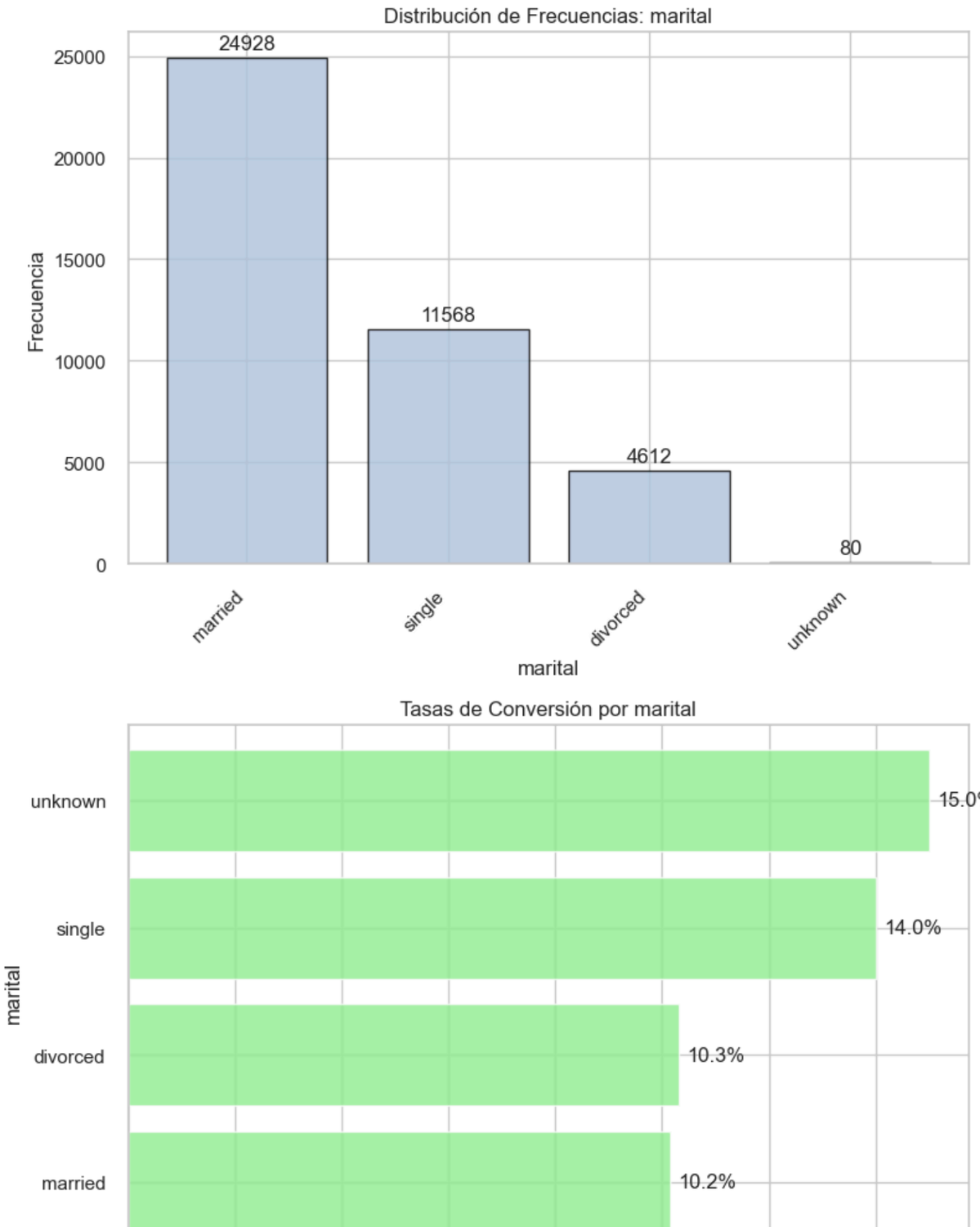


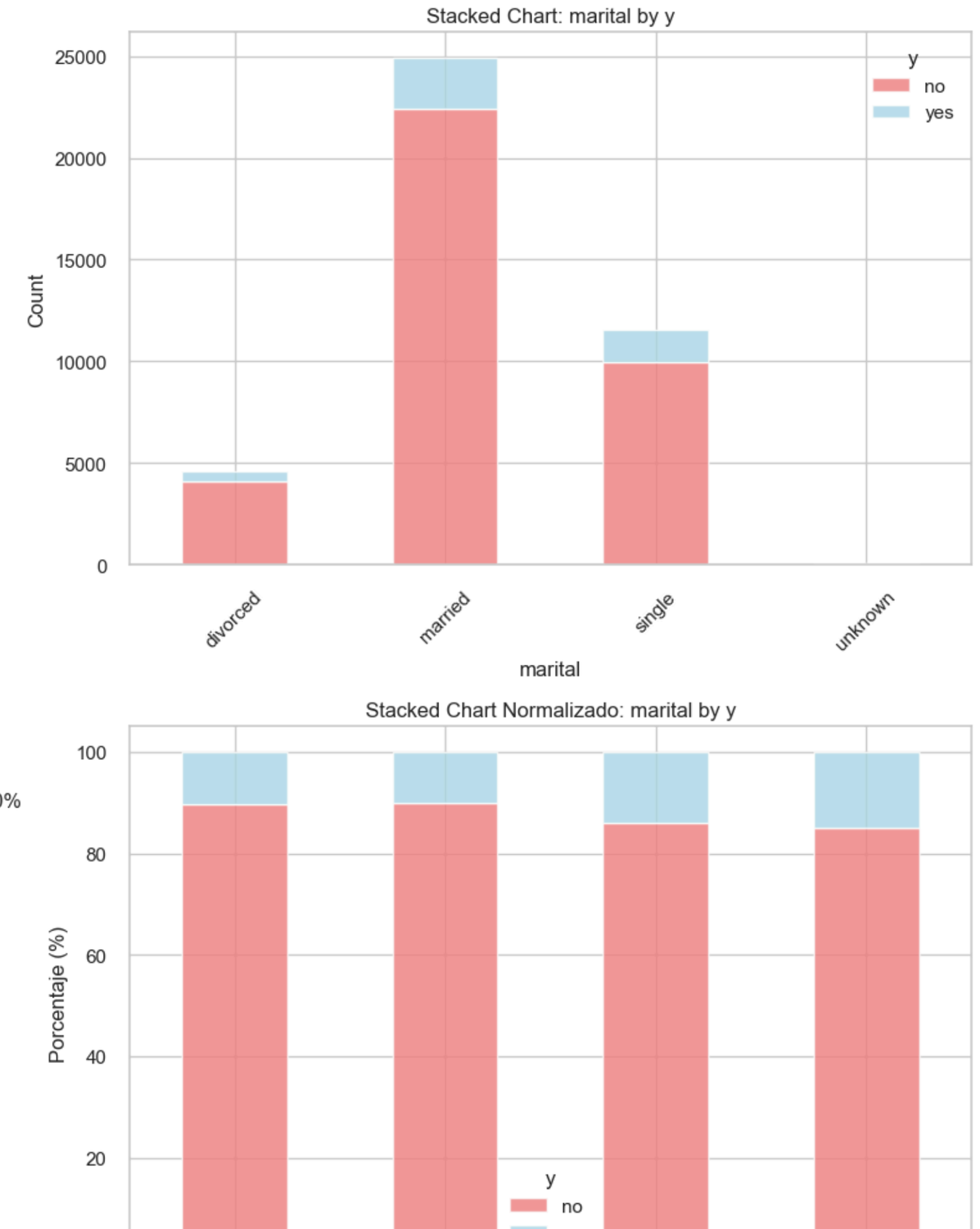



0
5
10
15
20
25
30
Tasa de Conversión (%)


0
admin.
blue-collar
entrepreneur
housemaid
management
retired
self-employed
services
student
technician
unemployed
unknown
job

GRÁFICOS DE BARRAS Y STACKED CHARTS (Variables Categóricas)

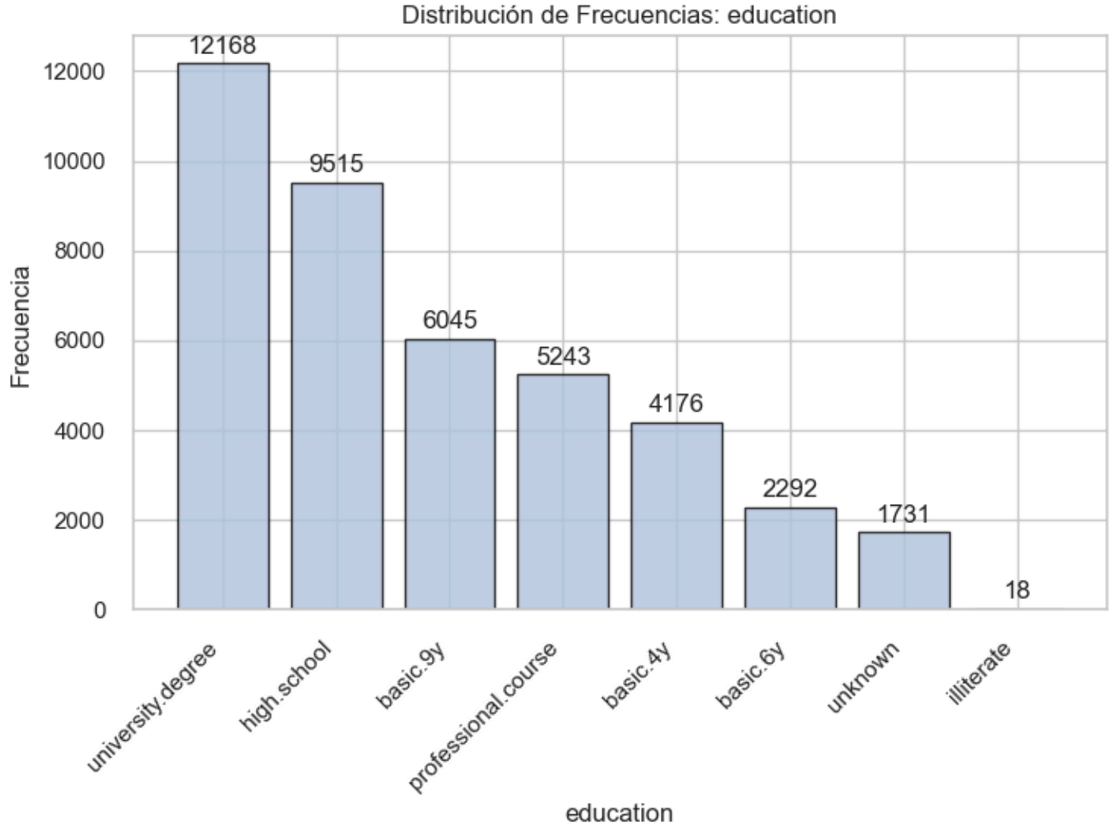


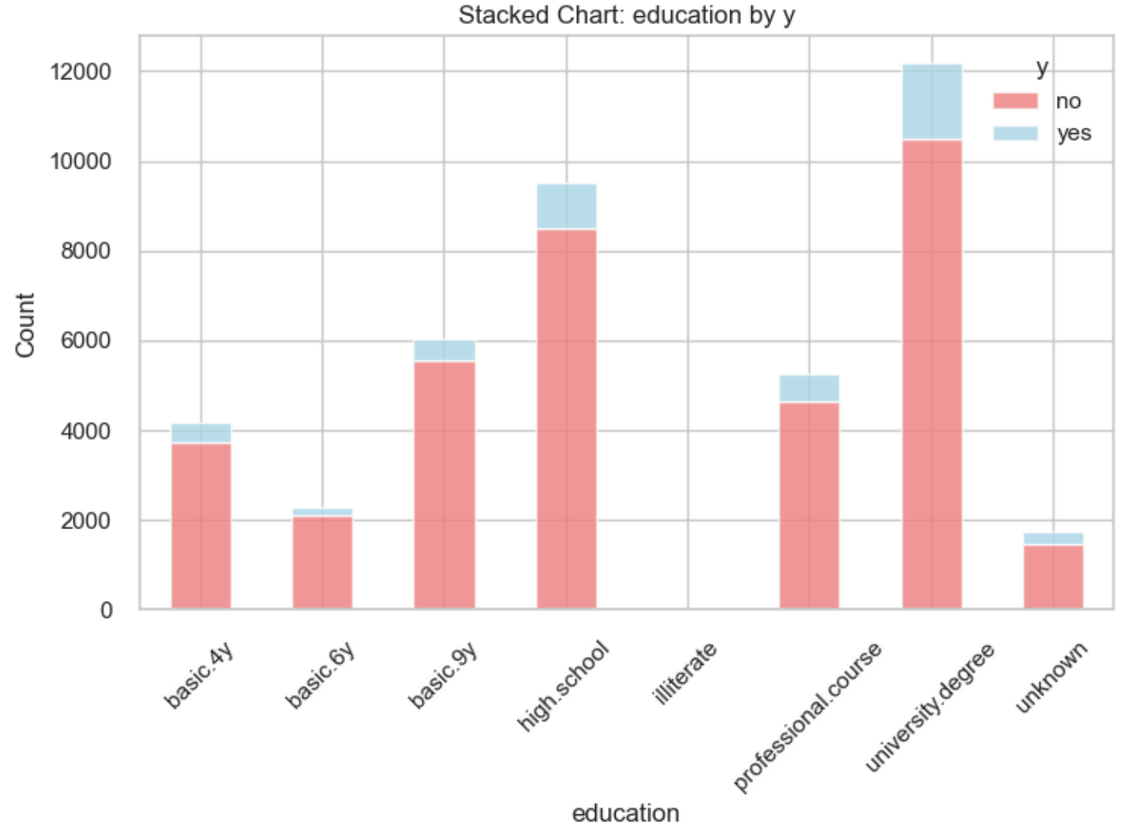


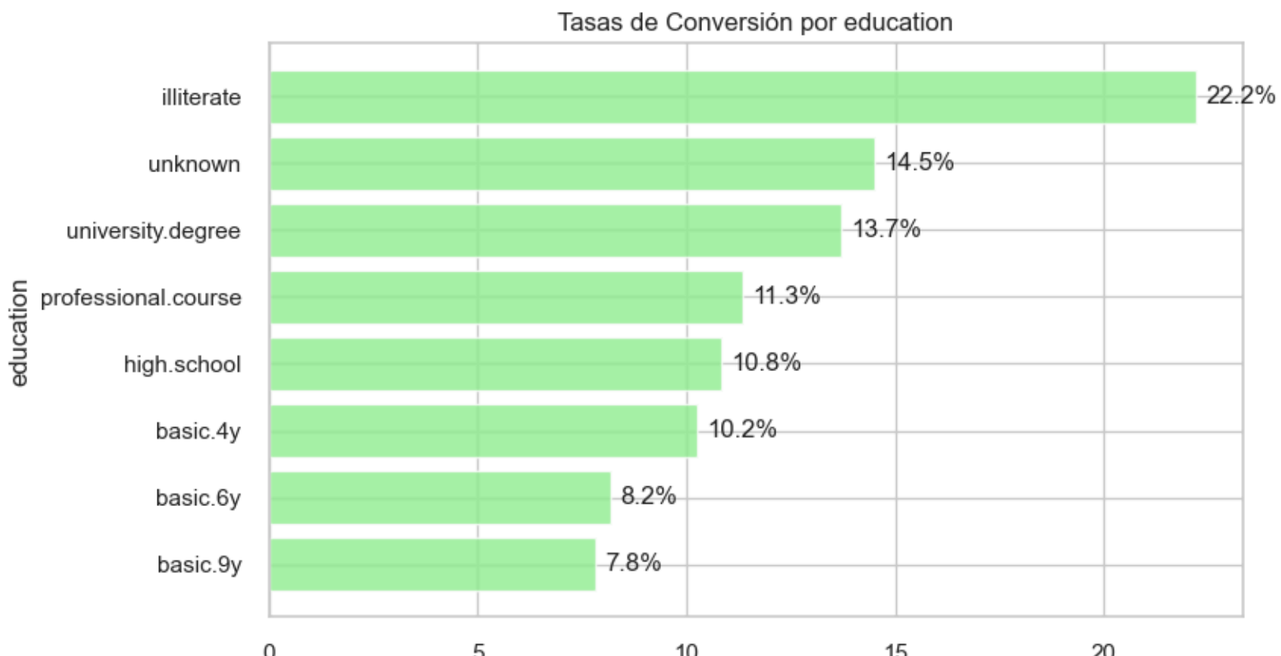

0
2
4
6
8
10
12
14
Tasa de Conversión (%)

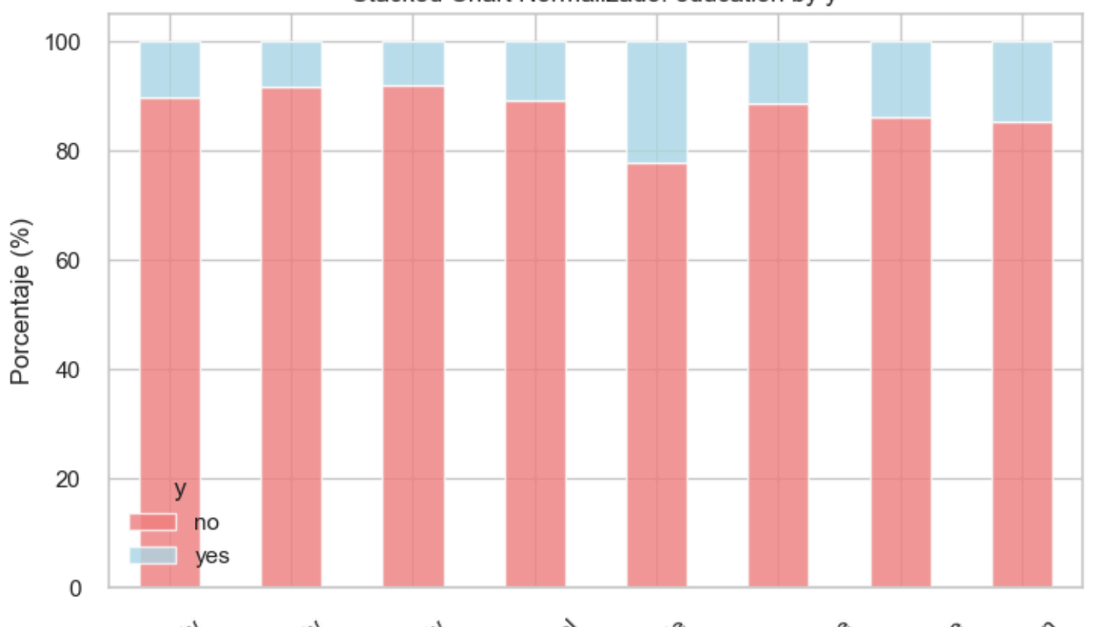

0
yes
divorced
married
single
unknown
marital

GRÁFICOS DE BARRAS Y STACKED CHARTS (Variables Categóricas)







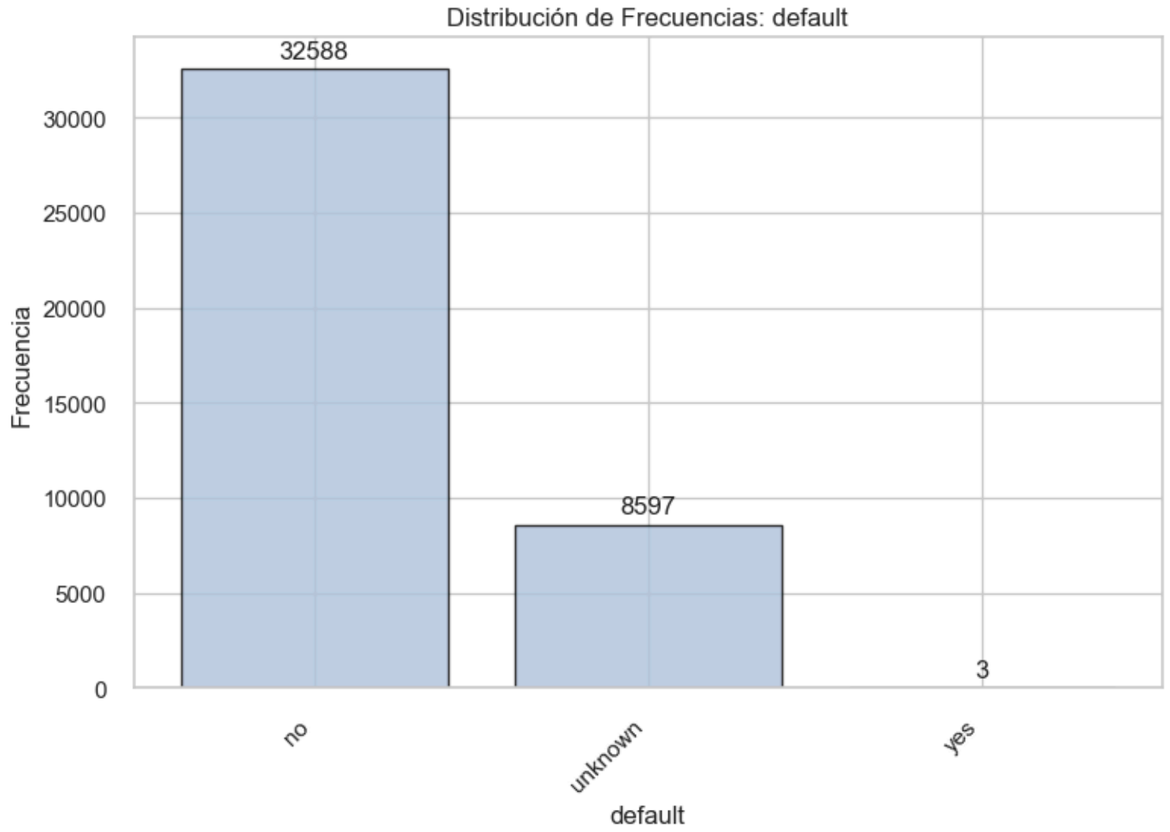


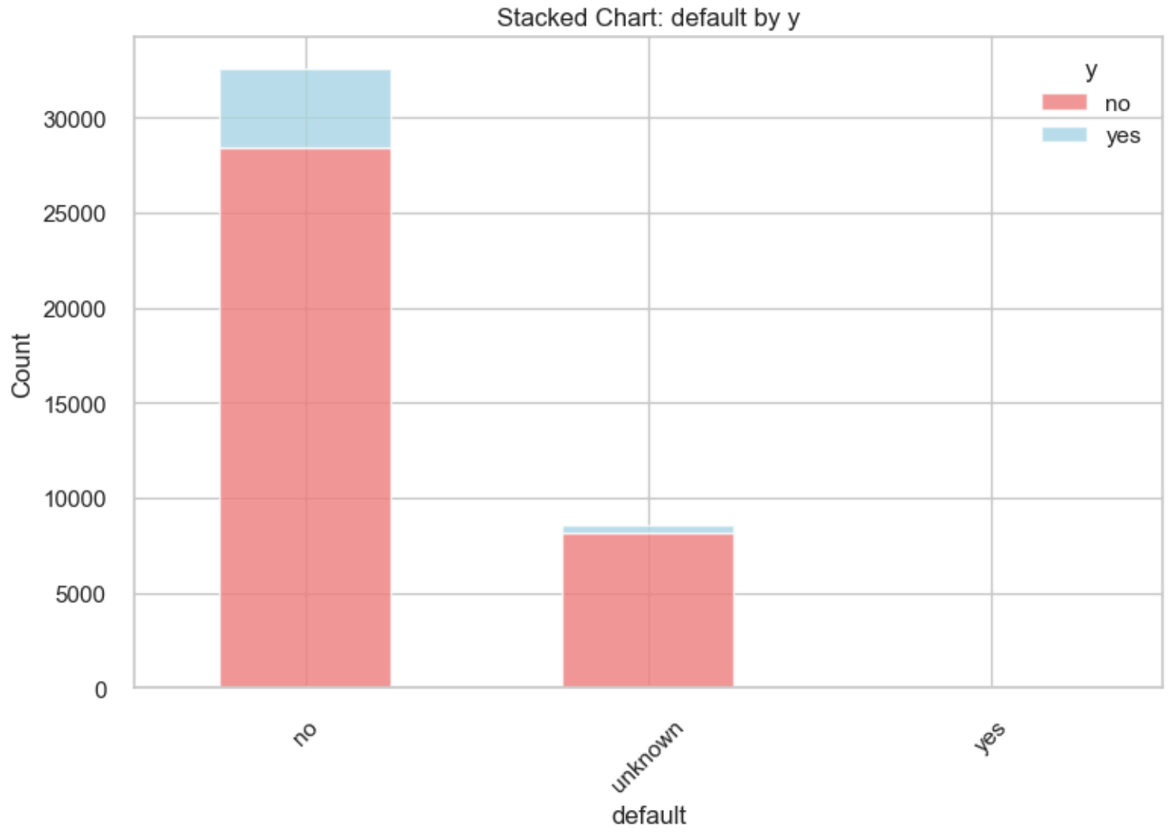
Tasa de Conversión (%)

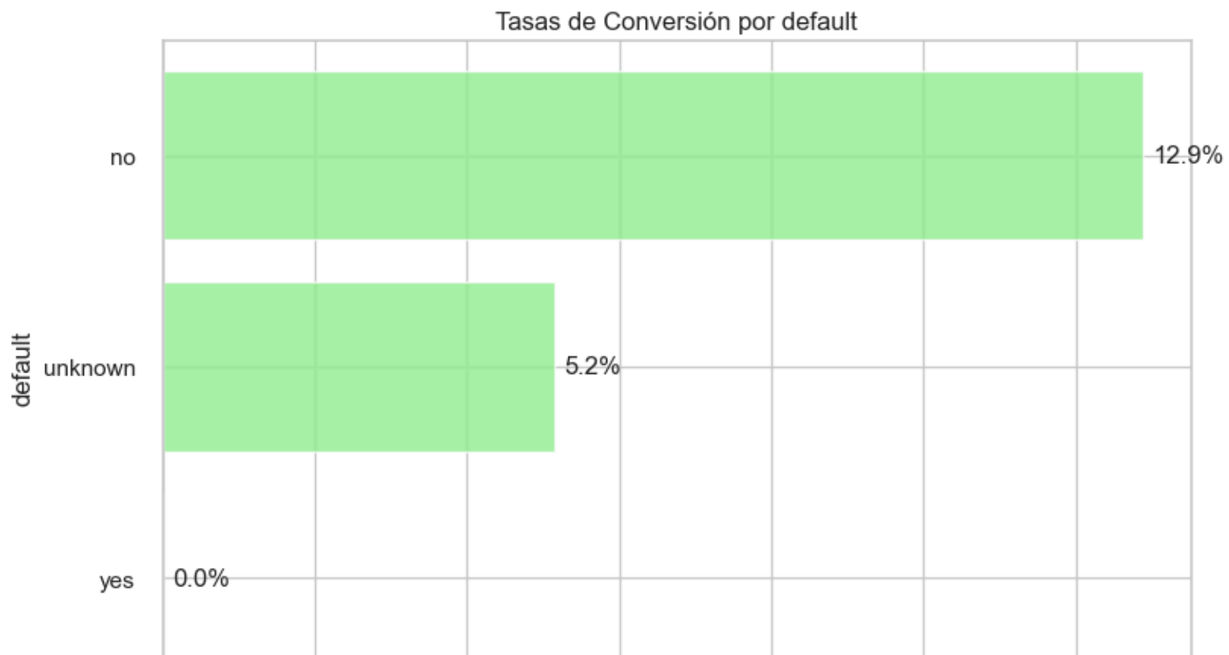
basic.4y
basic.6y
basic.9y
high.school
illiterate
professional.course
university.degree
unknown

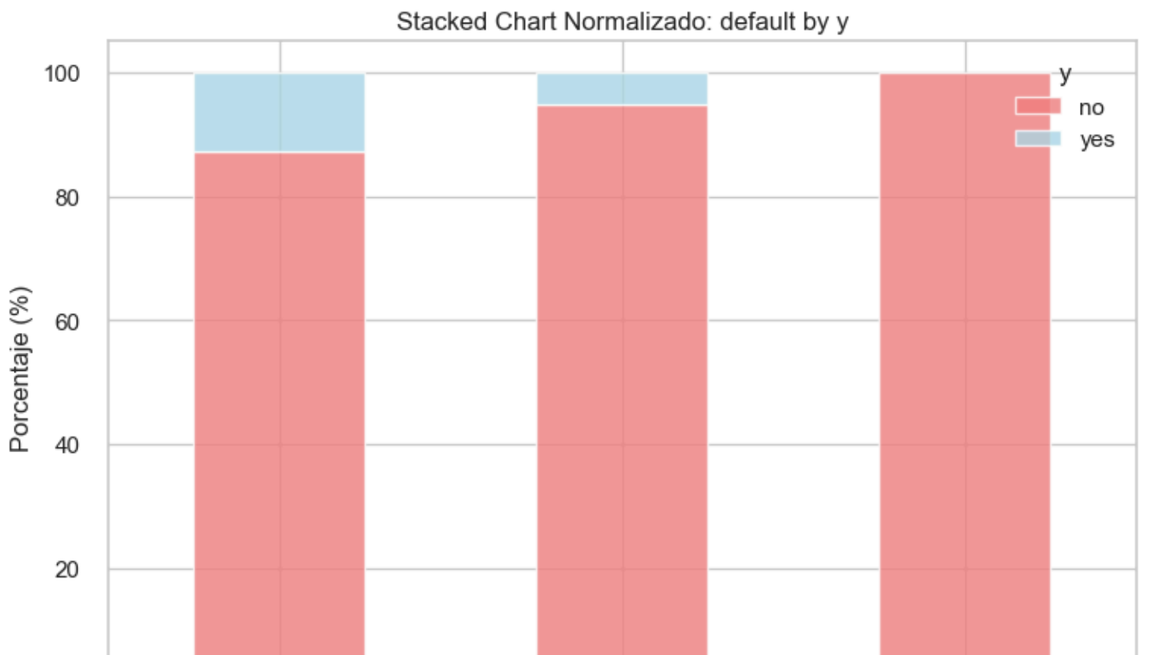
education

GRÁFICOS DE BARRAS Y STACKED CHARTS (Variables Categóricas)

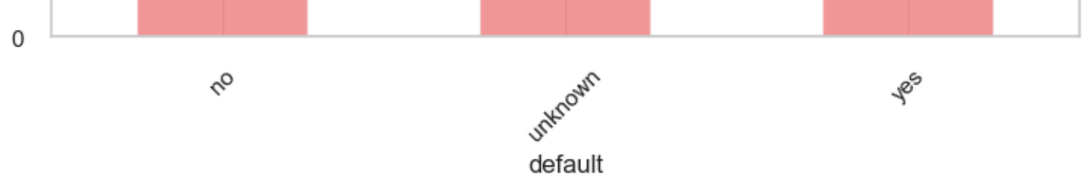




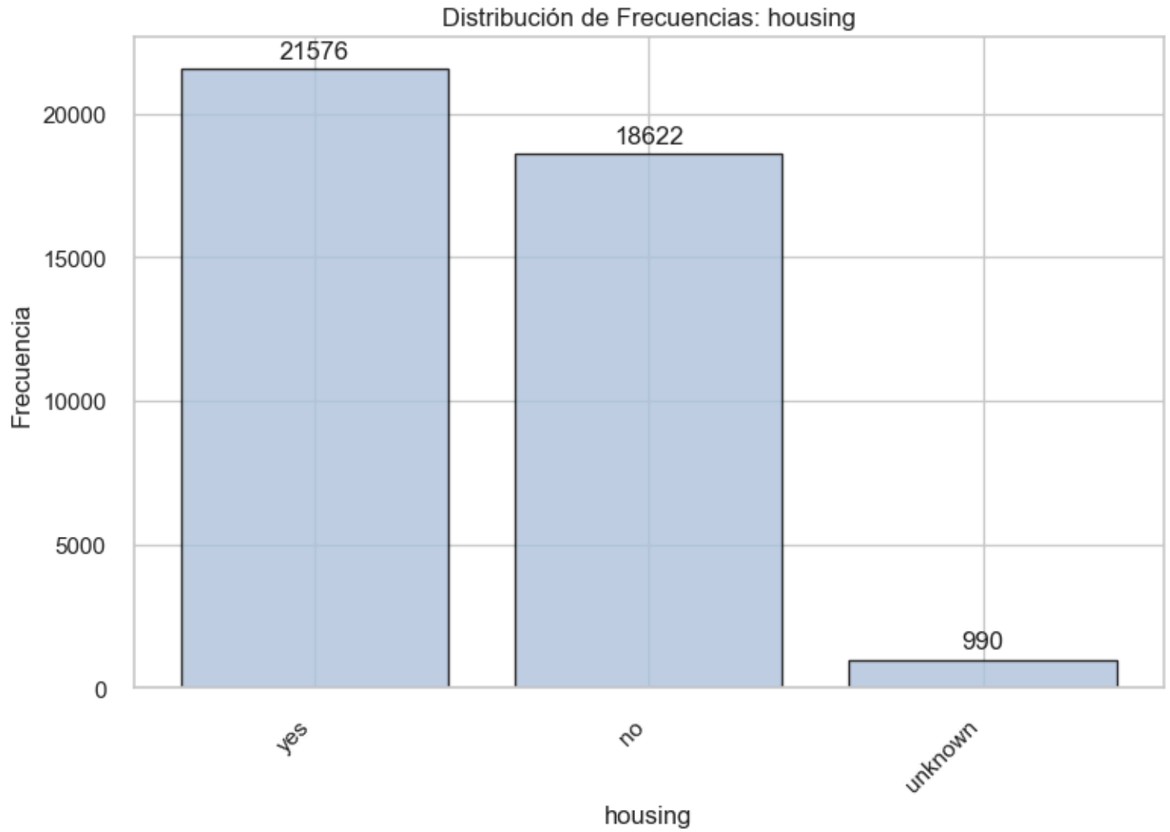


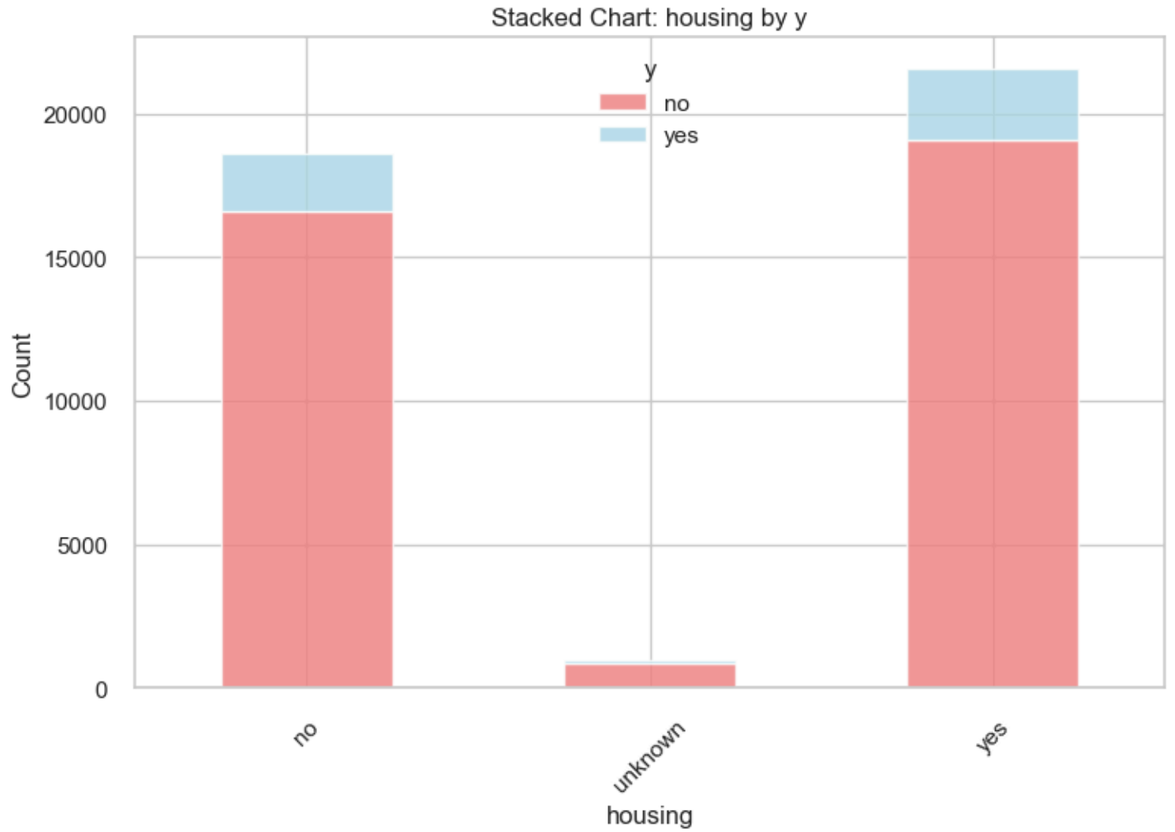


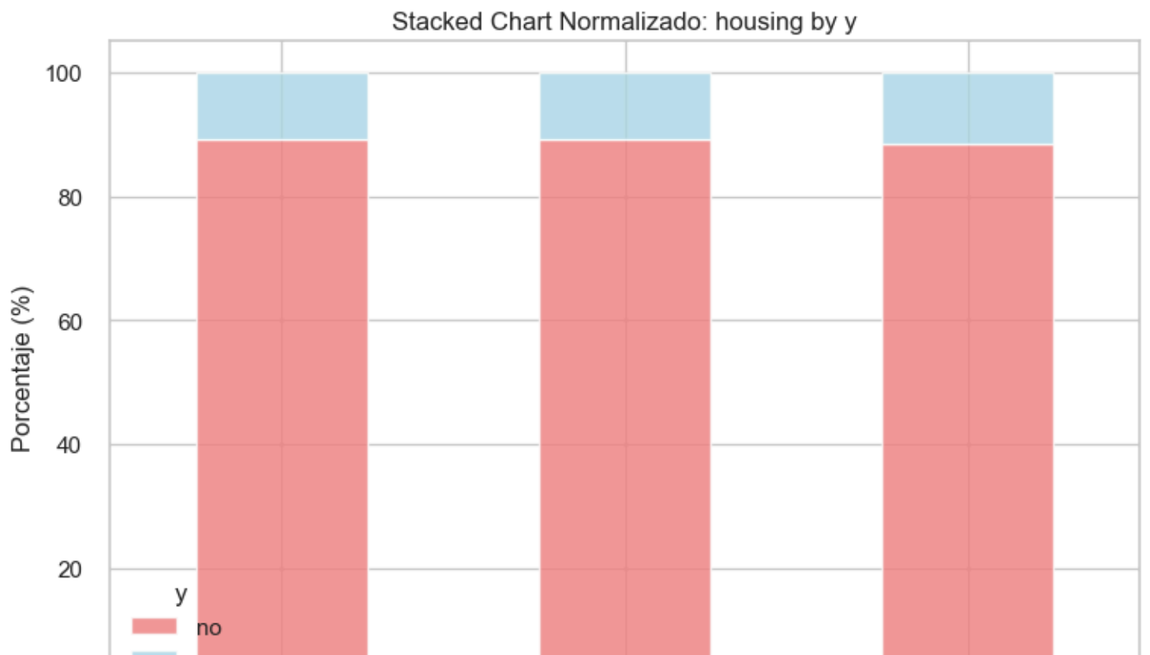

0
2
4
6
8
10
12
Tasa de Conversión (%)

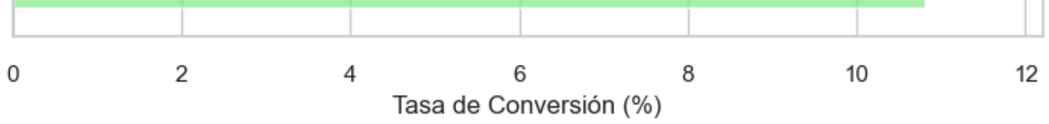

0
no
unknown
yes
default

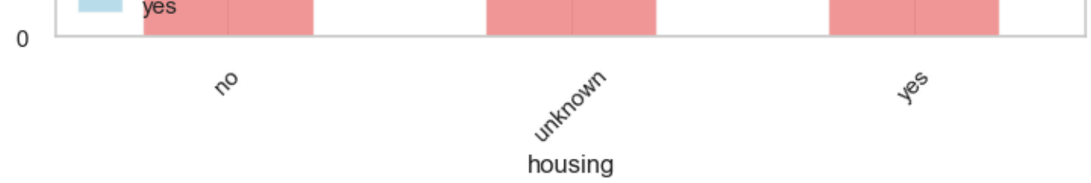
GRÁFICOS DE BARRAS Y STACKED CHARTS (Variables Categóricas)



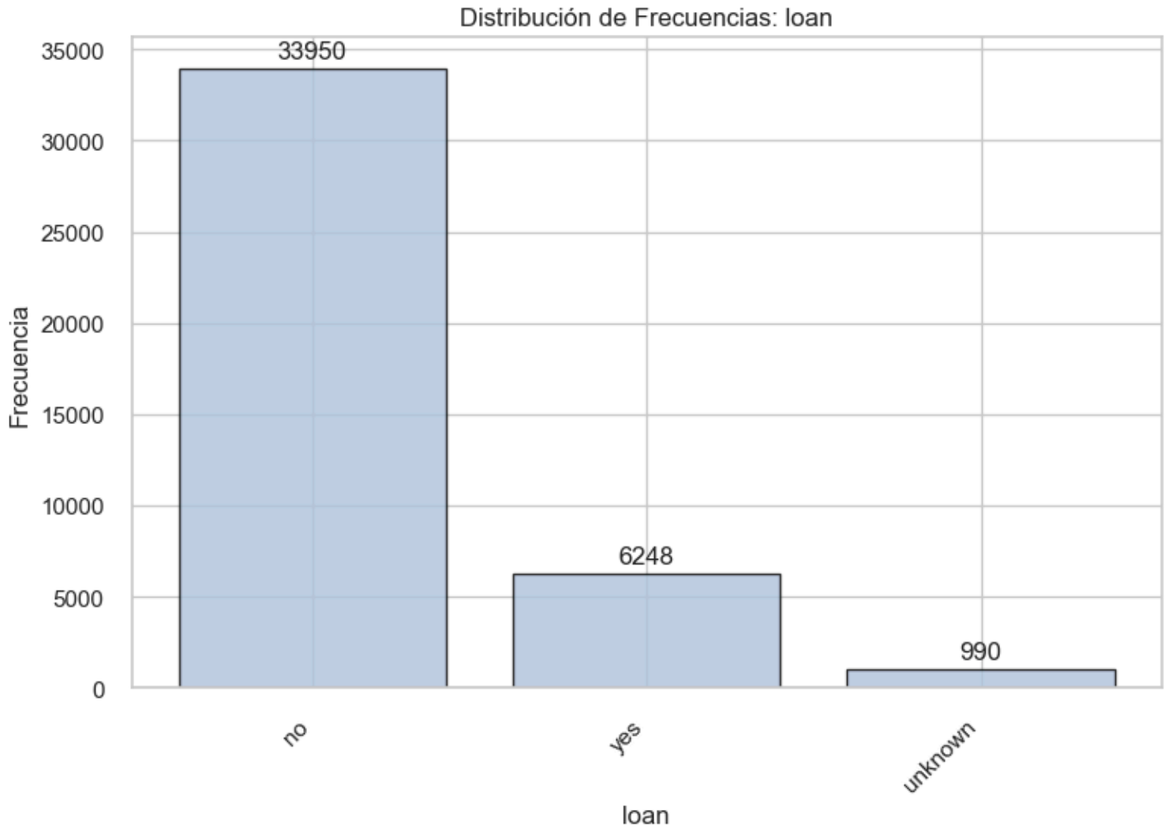


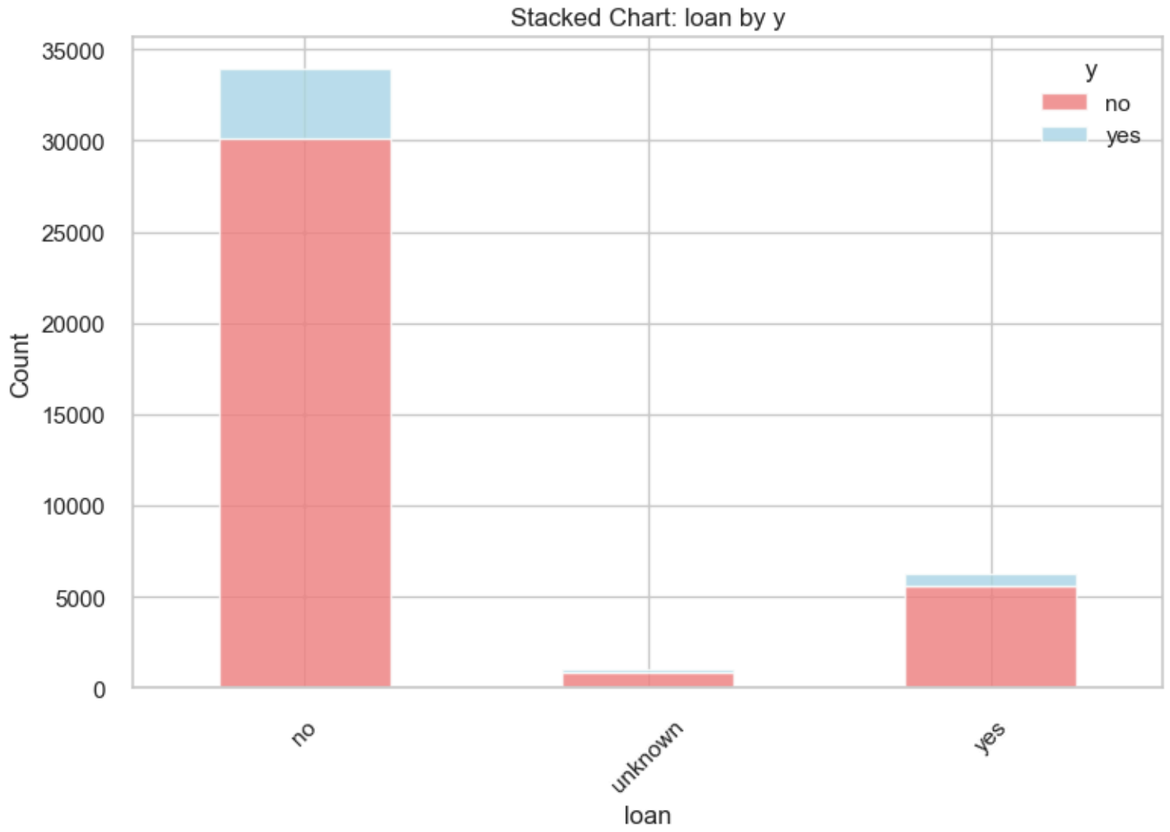


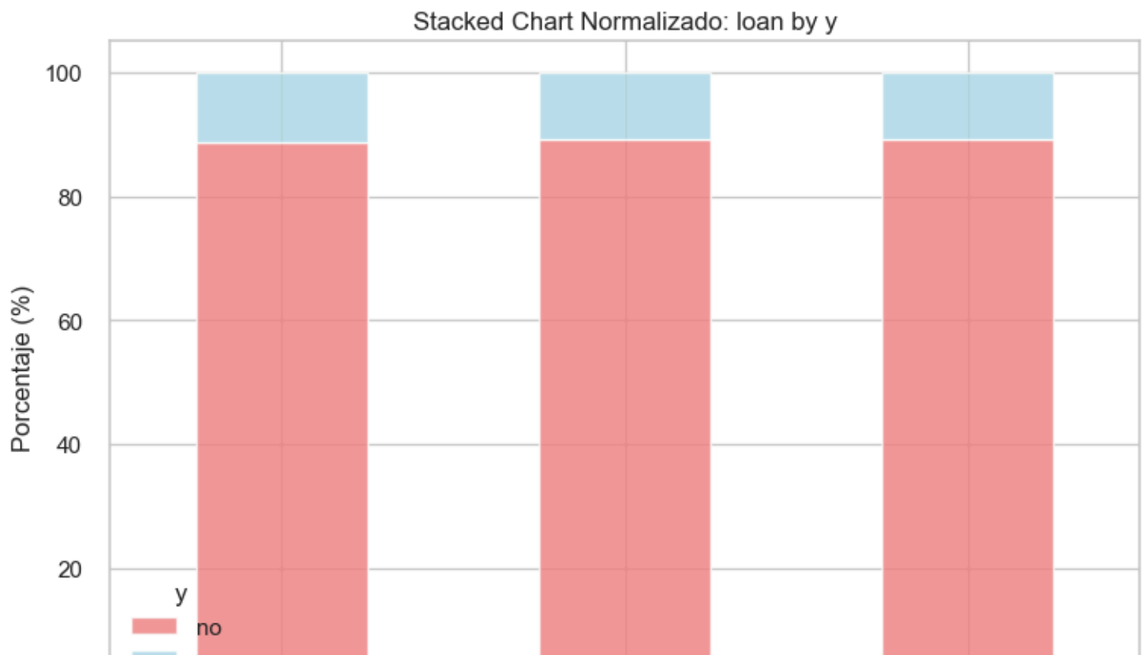

0
2
4
6
8
10
12
Tasa de Conversión (%)


yes
0
no
unknown
yes
housing

GRÁFICOS DE BARRAS Y STACKED CHARTS (Variables Categóricas)



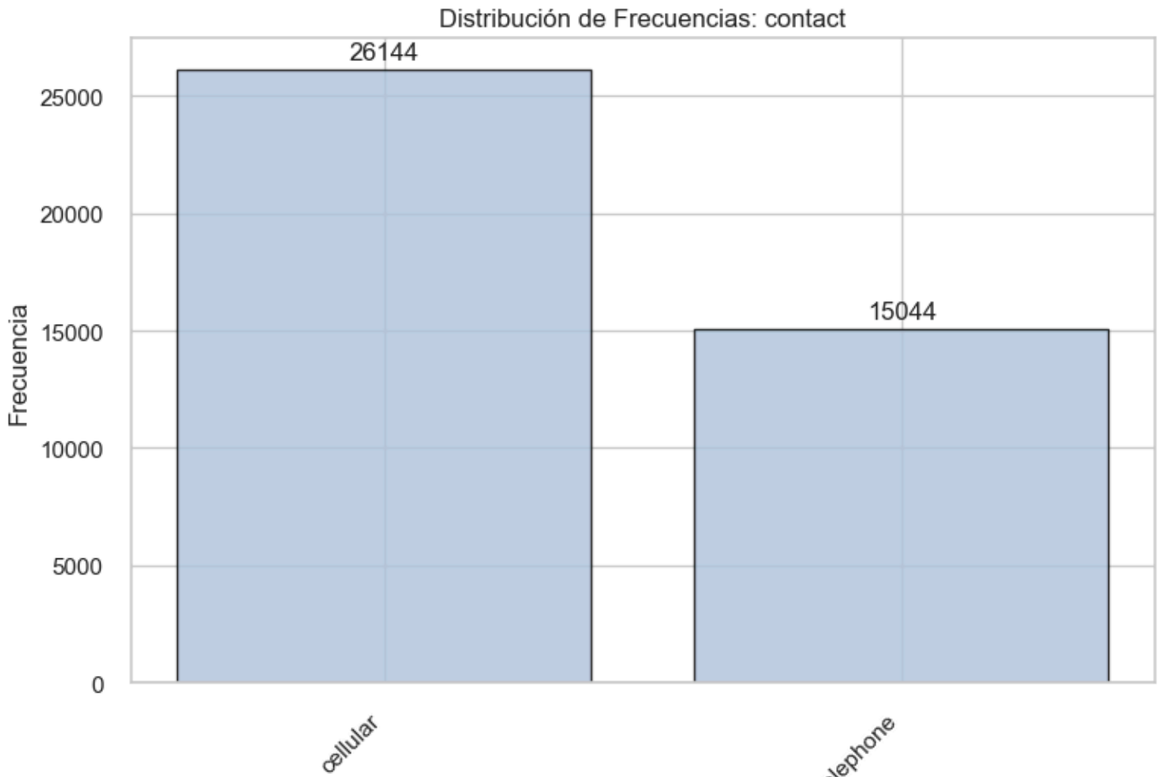


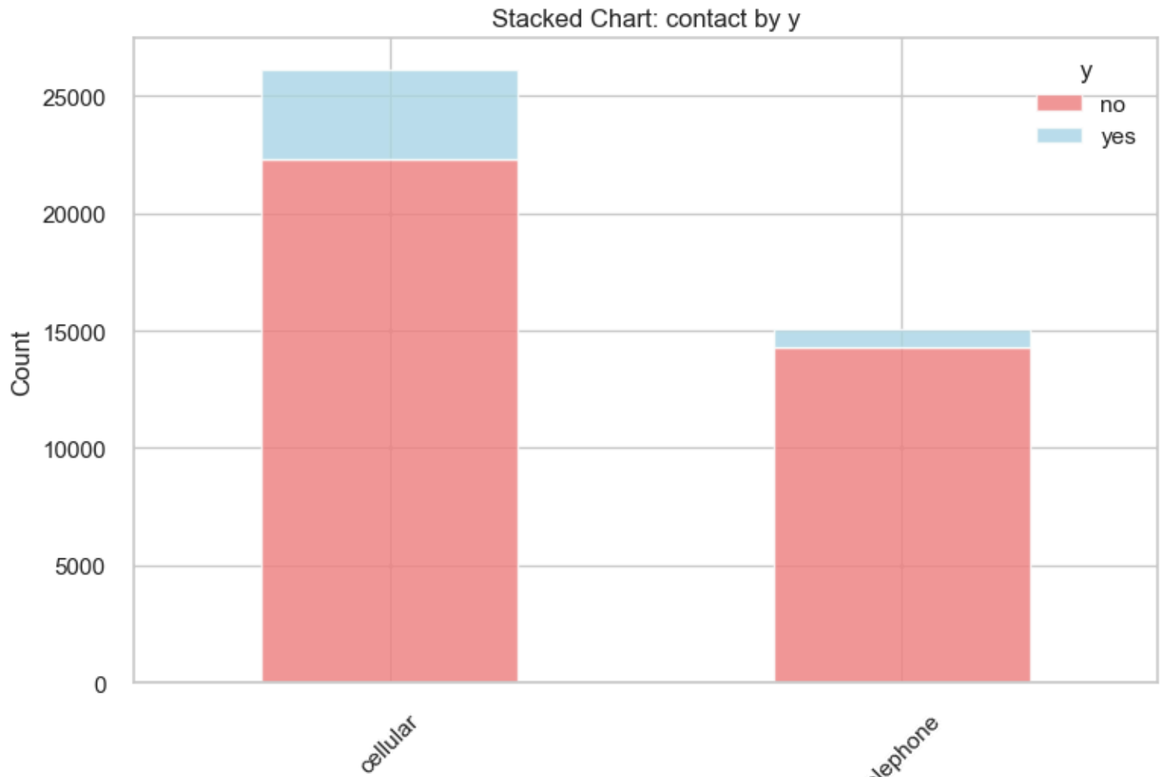


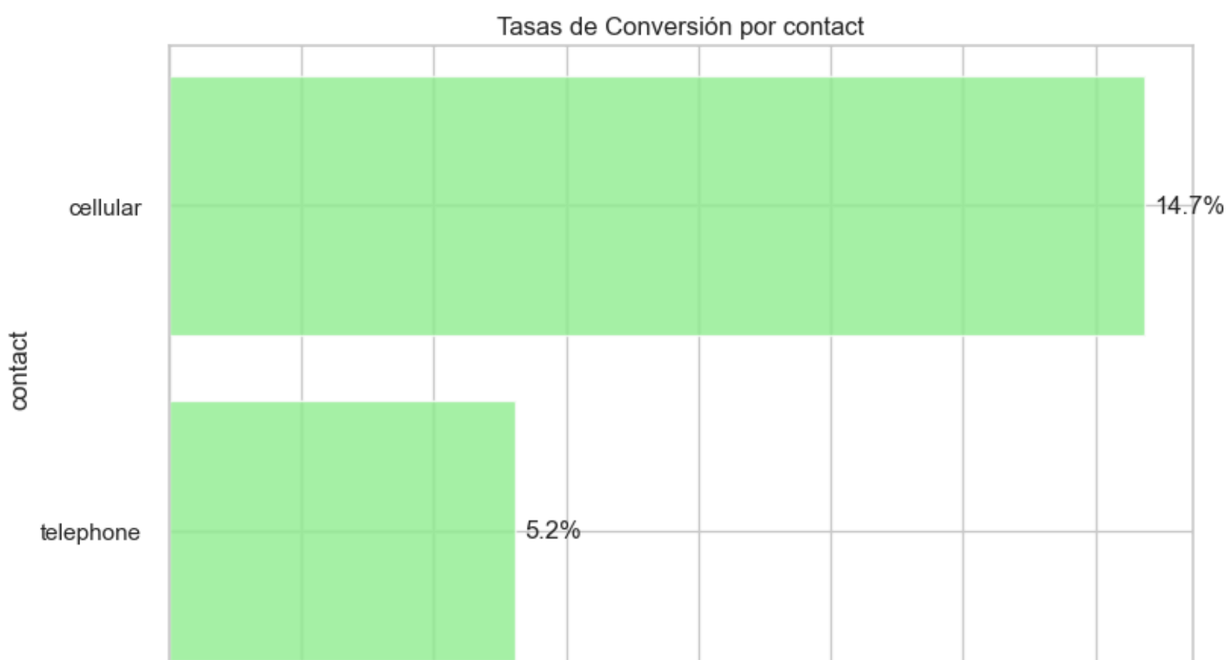

0
2
4
6
8
10
Tasa de Conversión (%)

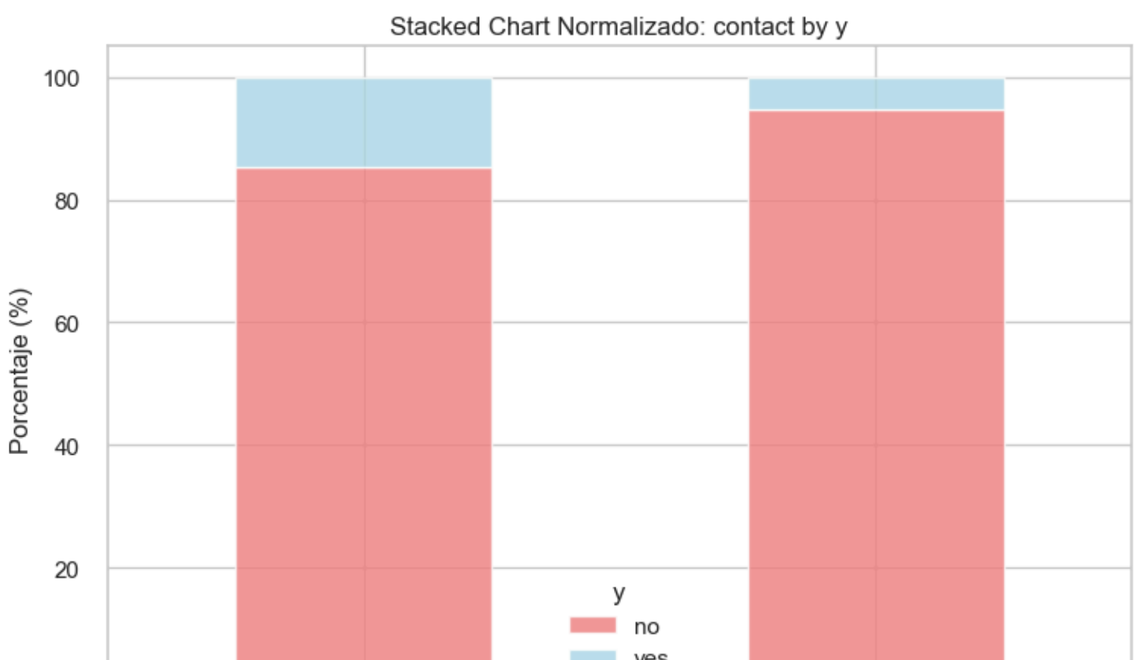

yes
0
no
unknown
yes
loan

GRÁFICOS DE BARRAS Y STACKED CHARTS (Variables Categóricas)

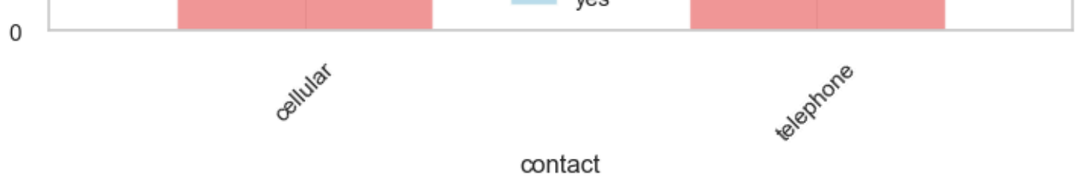




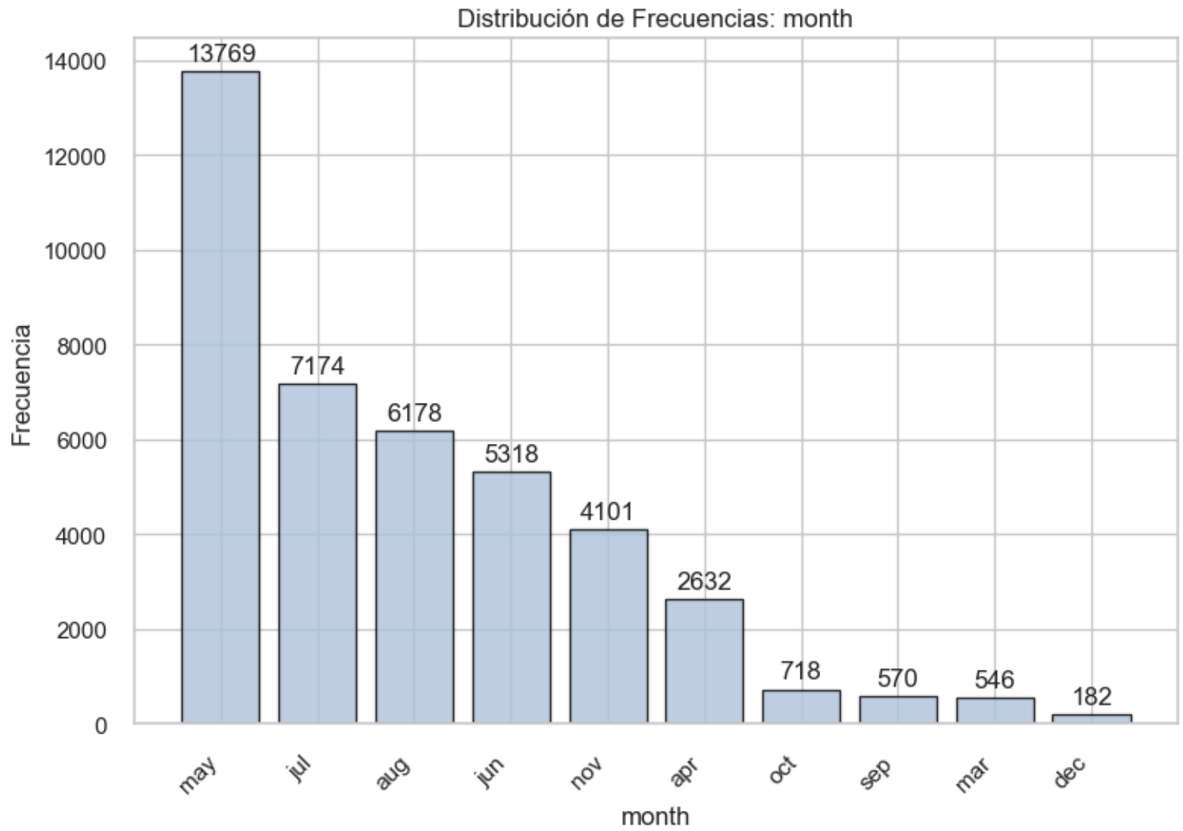


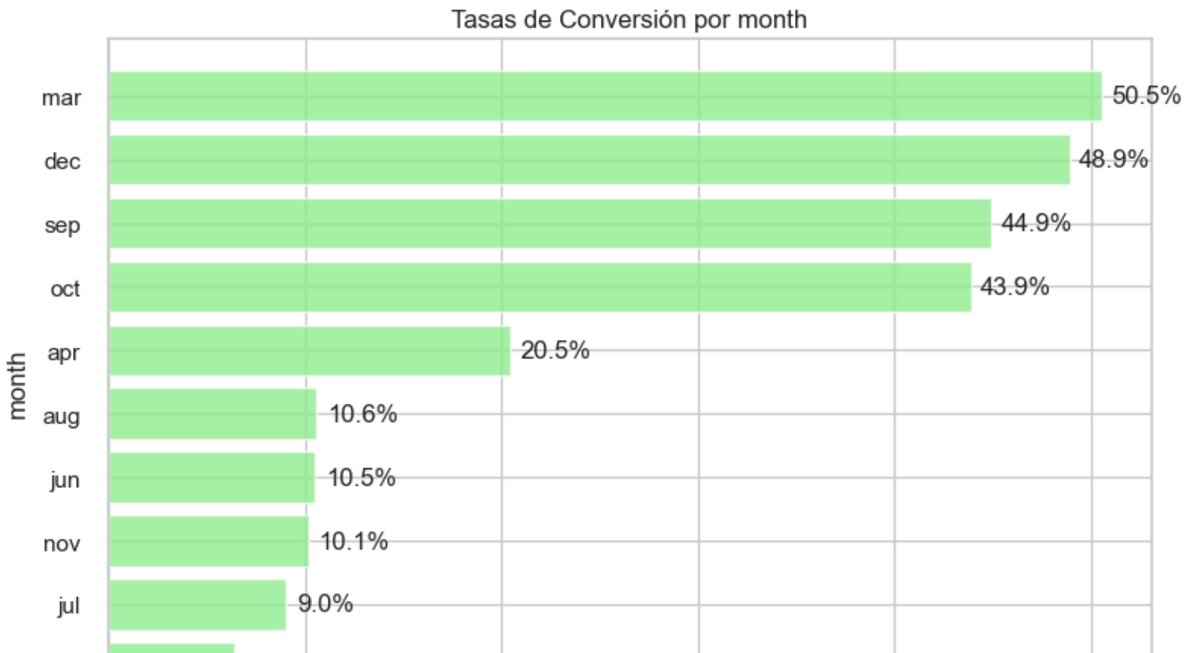


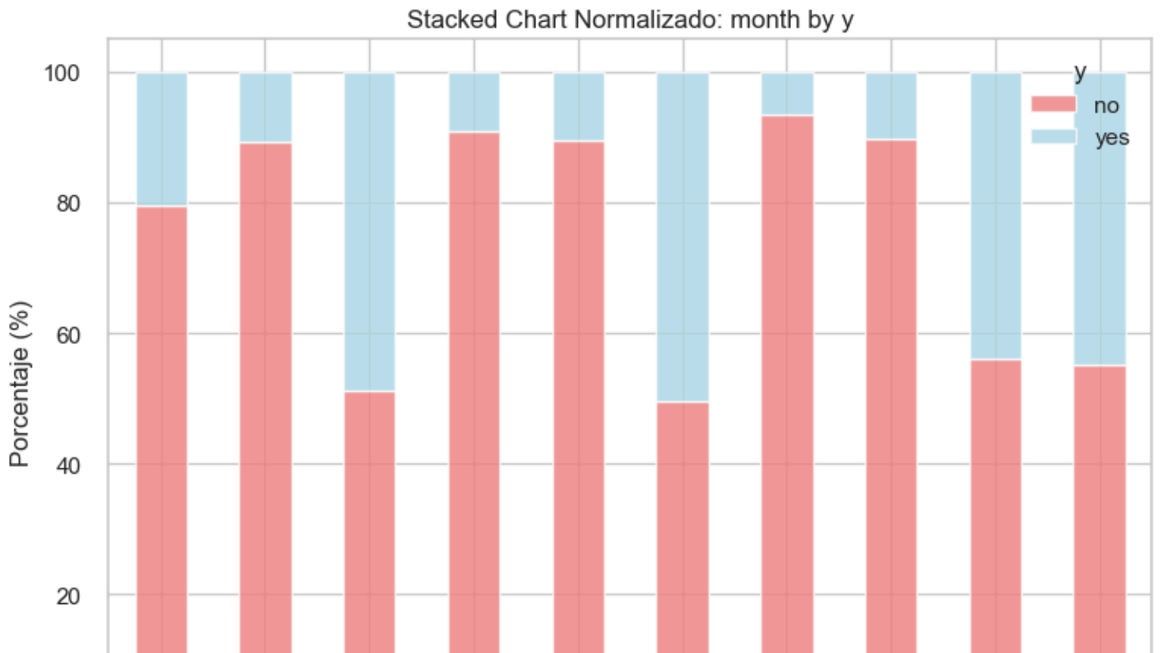

0
2
4
6
8
10
12
14
Tasa de Conversión (%)


0
yes
cellular
telephone
contact

GRÁFICOS DE BARRAS Y STACKED CHARTS (Variables Categóricas)



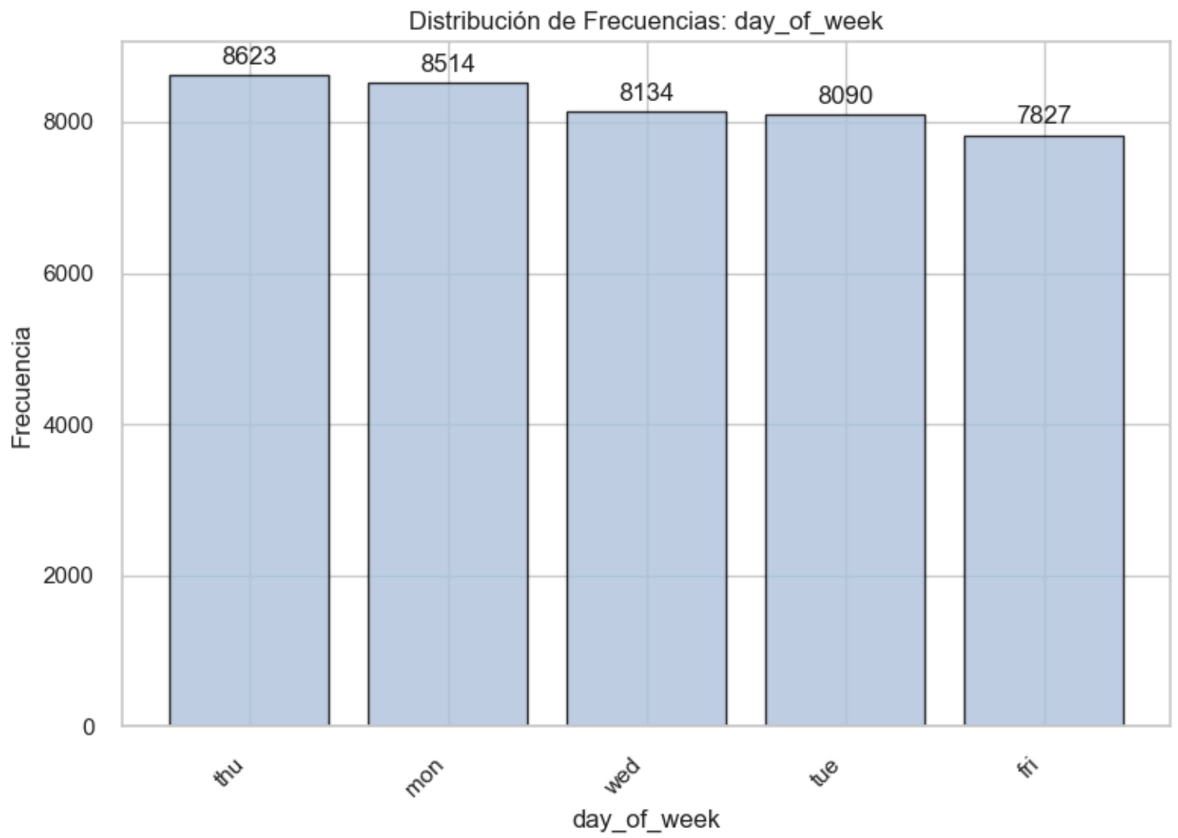


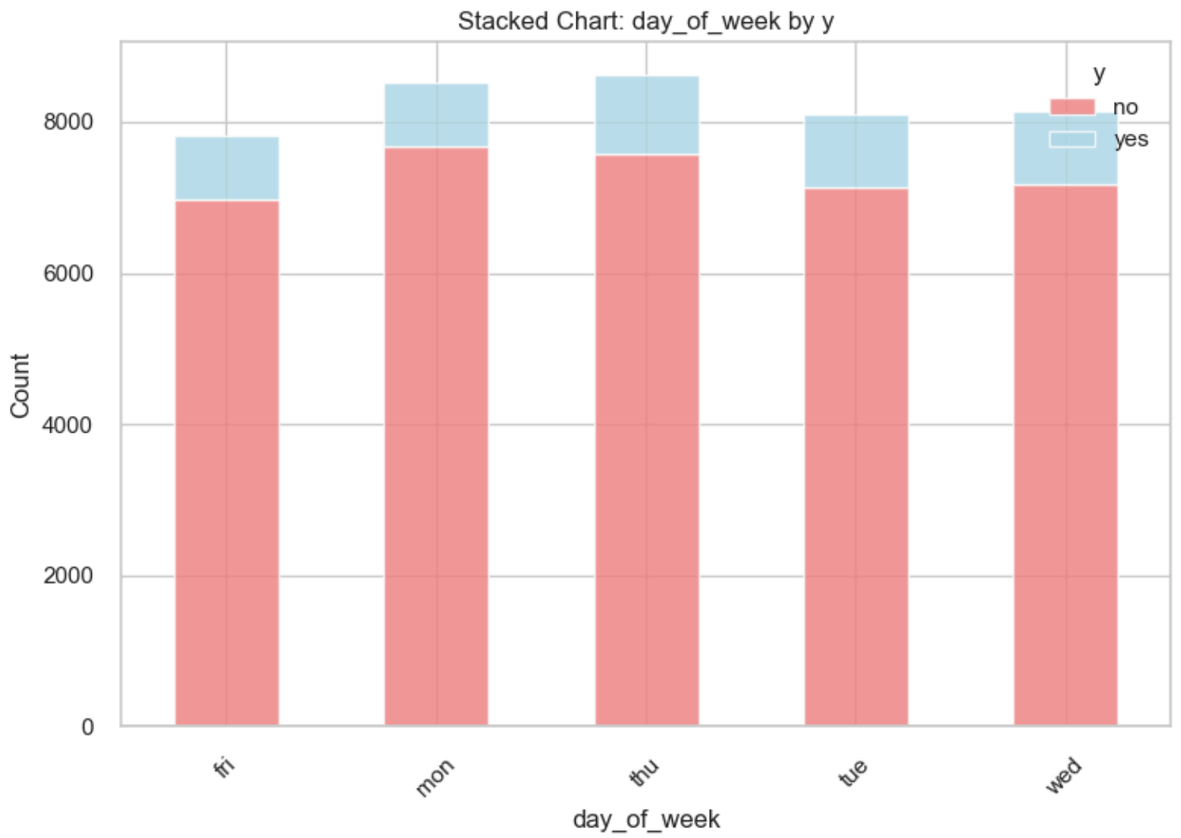


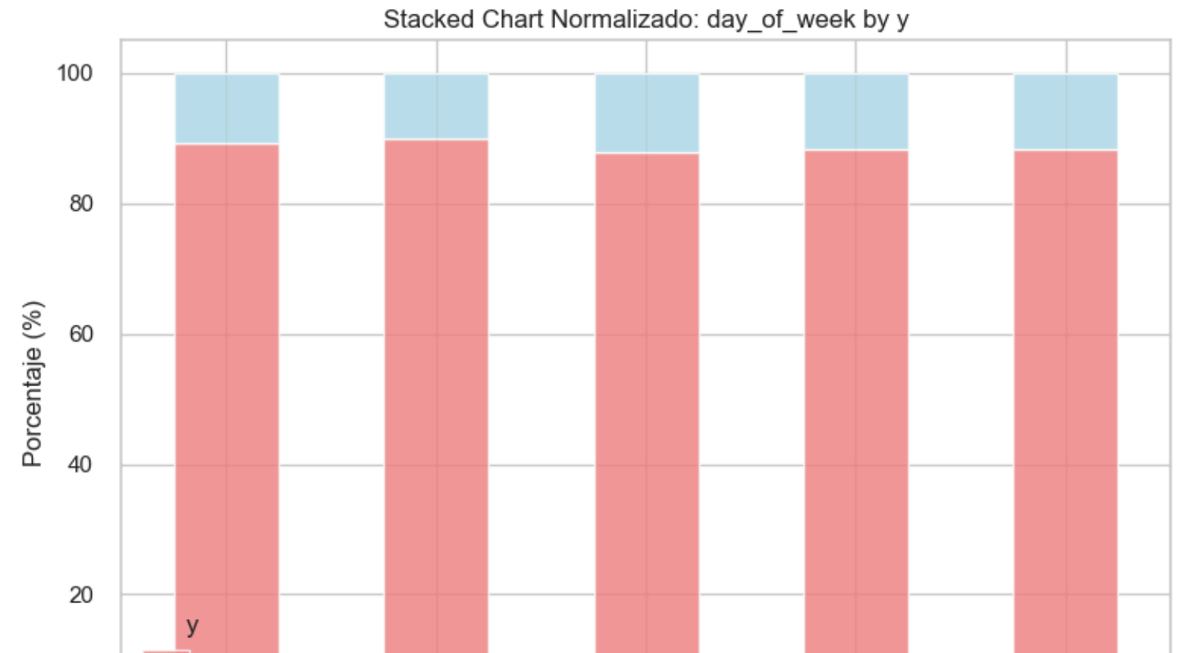

may
6.4%
0
10
20
30
40
50
Tasa de Conversión (%)

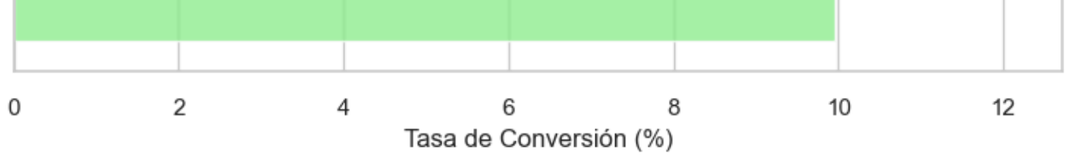

0
apr
aug
dec
jul
jun
mar
may
nov
oct
sep
month

GRÁFICOS DE BARRAS Y STACKED CHARTS (Variables Categóricas)



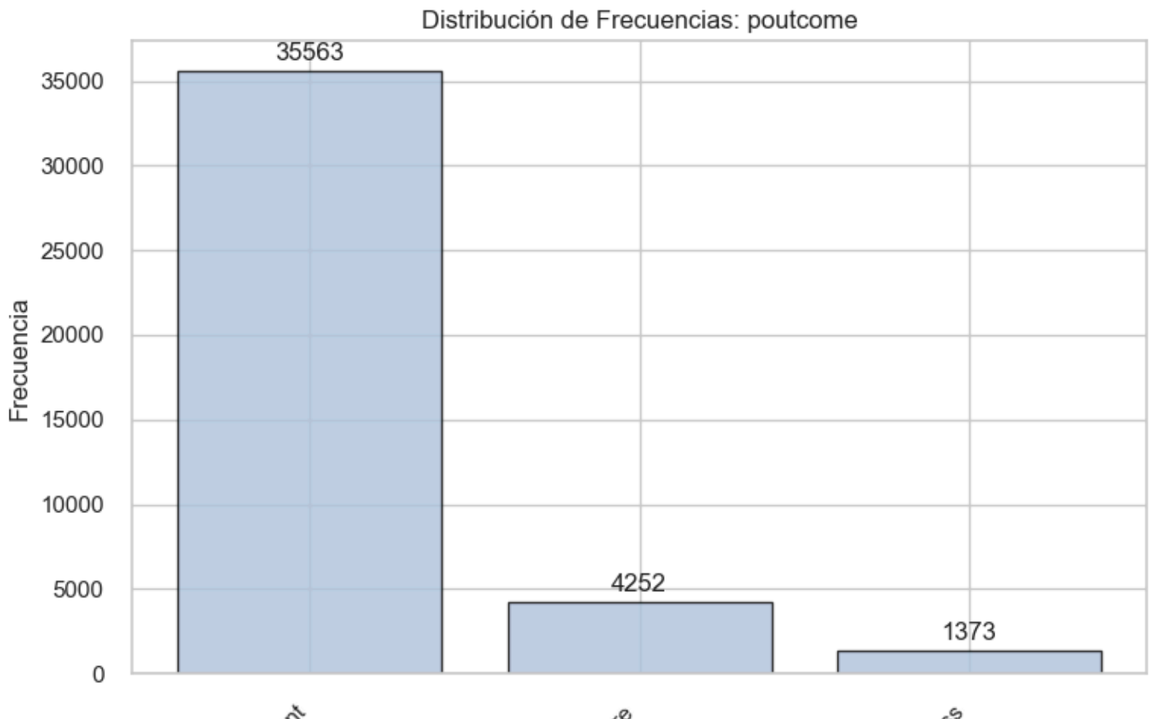


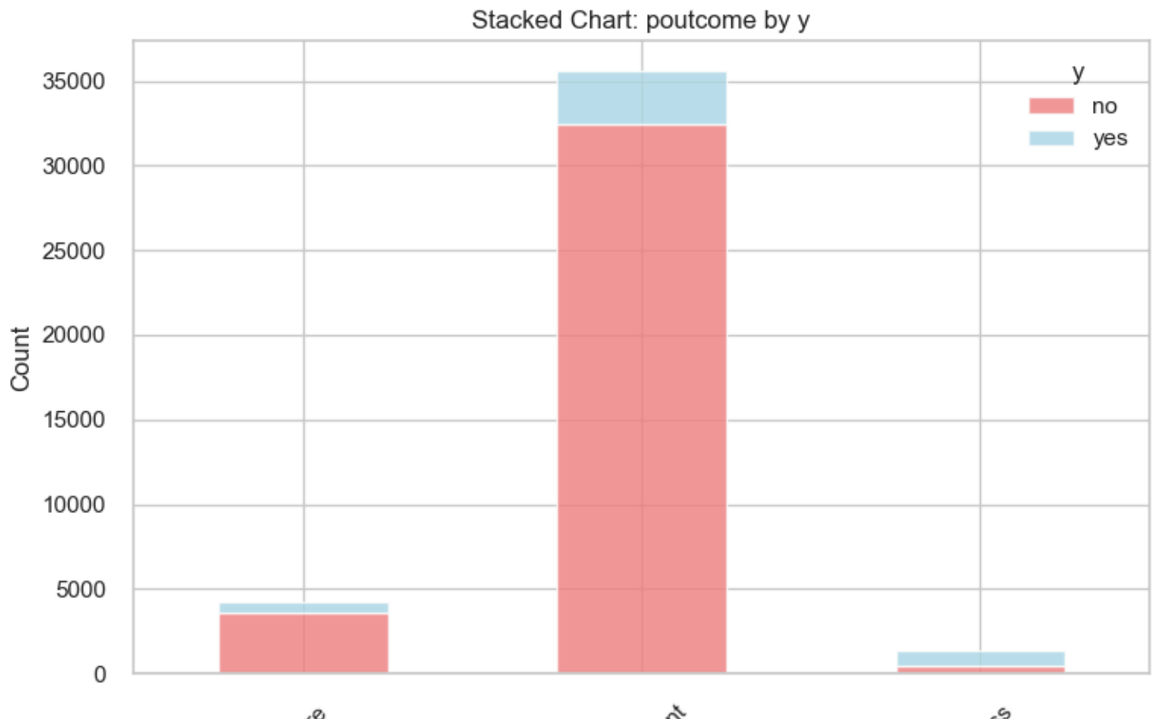


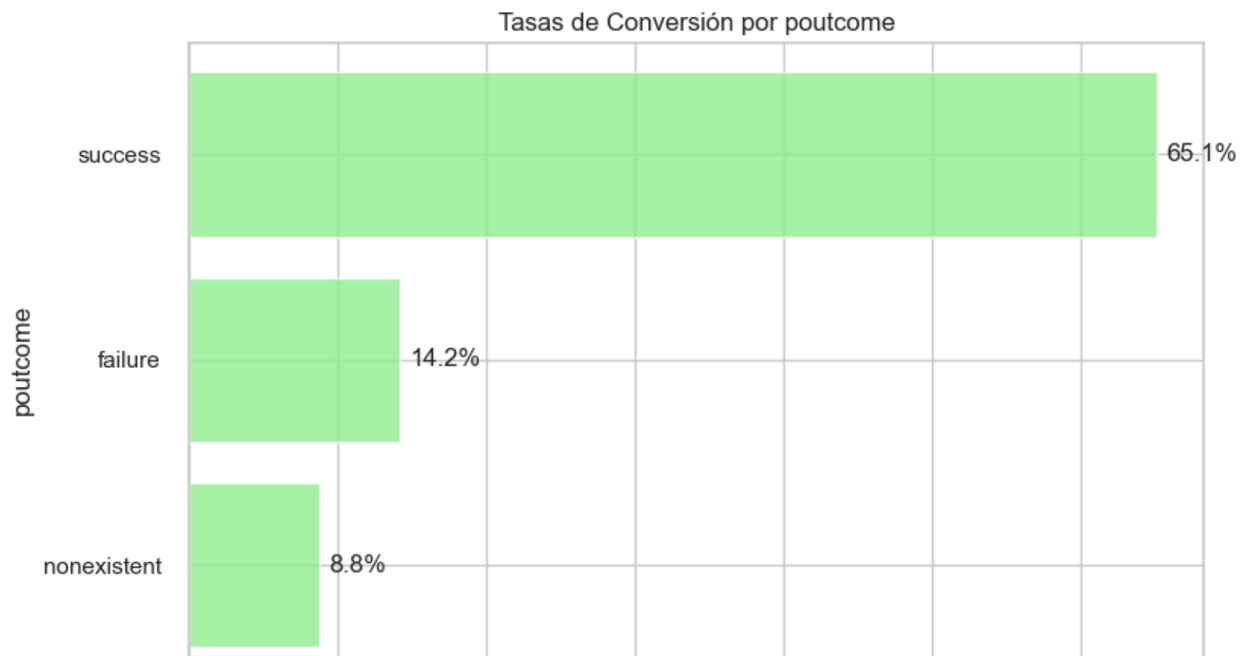

0
2
4
6
8
10
12
Tasa de Conversión (%)

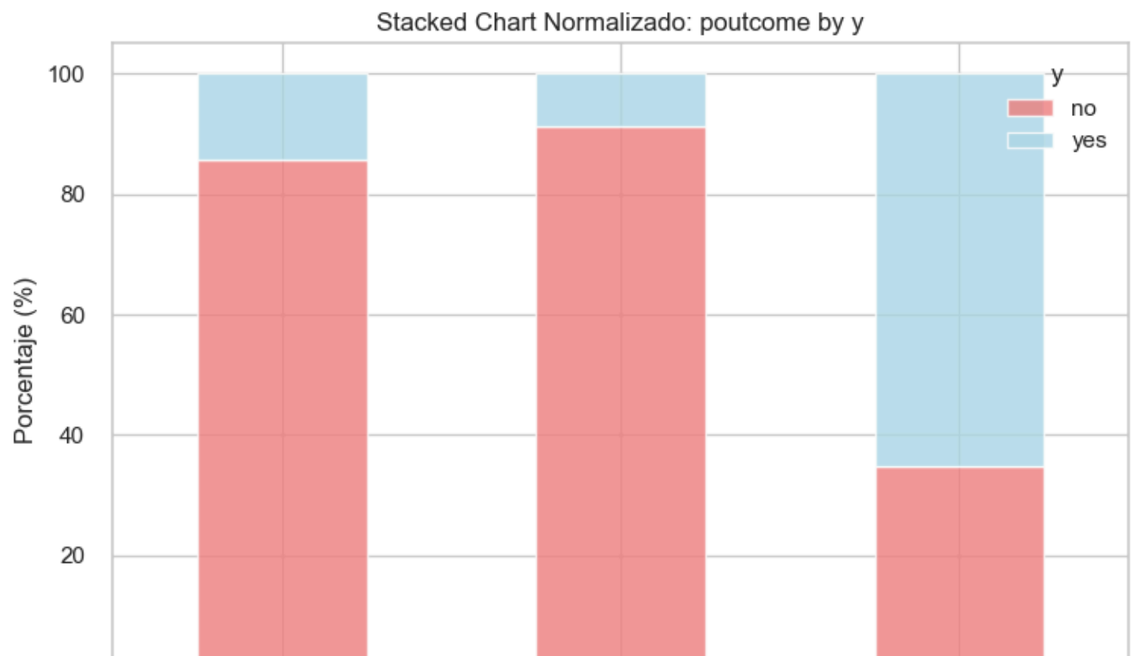

no
yes
0
fri
mon
thu
tue
wed
day_of_week

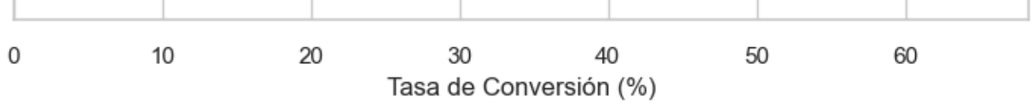
GRÁFICOS DE BARRAS Y STACKED CHARTS (Variables Categóricas)











In [18]:
```
# Step 9: Generate reports
print("Step 9: Generating reports...")
generate_data_quality_report(quality_analysis, var_types)
generate_target_analysis_report(target_analysis)
generate_bivariate_report(numeric_bivariate, categorical_bivariate,
                          selected_numeric, selected_categorical)
generate_summary_report(basic_info, target_analysis, correlation_analysis)
```

```
Step 9: Generating reports...

DATA QUALITY REPORT
==================
Dataset Shape: (22, 6)
Total Memory Usage: 30987.2 KB

Variable Types:
  Numeric: 10 variables
  Categorical: 11 variables

Low Variability Variables:
  default: 3 unique values
  housing: 3 unique values
  loan: 3 unique values
  contact: 2 unique values
  poutcome: 3 unique values
  y: 2 unique values

Data Quality Summary:
                  dtype  non_null  null_count  null_percentage  unique_values  \
Index               NaN       NaN         NaN              NaN            NaN
age               int64   41188.0         0.0              0.0           78.0
campaign          int64   41188.0         0.0              0.0           42.0
cons.conf.idx   float64   41188.0         0.0              0.0           26.0
cons.price.idx  float64   41188.0         0.0              0.0           26.0
contact          object   41188.0         0.0              0.0            2.0
day_of_week      object   41188.0         0.0              0.0            5.0
default          object   41188.0         0.0              0.0            3.0
duration          int64   41188.0         0.0              0.0         1544.0
education        object   41188.0         0.0              0.0            8.0
emp.var.rate    float64   41188.0         0.0              0.0           10.0
euribor3m       float64   41188.0         0.0              0.0          316.0
housing          object   41188.0         0.0              0.0            3.0
job              object   41188.0         0.0              0.0           12.0
loan             object   41188.0         0.0              0.0            3.0
marital          object   41188.0         0.0              0.0            4.0
month            object   41188.0         0.0              0.0           10.0
nr.employed     float64   41188.0         0.0              0.0           11.0
pdays             int64   41188.0         0.0              0.0           27.0
poutcome         object   41188.0         0.0              0.0            3.0
previous          int64   41188.0         0.0              0.0            8.0
y                object   41188.0         0.0              0.0            2.0

                memory_usage
Index                    128
age                   329504
campaign              329504
cons.conf.idx         329504
cons.price.idx        329504
contact              2692264
day_of_week          2471280
default              2473080
duration              329504
education            2871255
emp.var.rate          329504
euribor3m             329504
housing              2456618
job                  2716564
loan                 2441290
marital              2629076
month                2471280
```

```
nr.employed           329504
pdays                 329504
poutcome             2778284
previous              329504
y                    2434732

TARGET VARIABLE ANALYSIS
========================
Distribution:
  no: 36548 (88.73%)
  yes: 4640 (11.27%)

Dataset Balance: Imbalanced
Imbalance Ratio: 7.9:1
Minority Class: yes

BIVARIATE ANALYSIS REPORT
=========================

DURATION vs TARGET:
Group Statistics:
     count    mean  median     std  min   max
y
no   36548  220.84   163.5  207.10    0  4918
yes   4640  553.19   449.0  401.17   37  4199

Statistical Test (t-test):
  t-statistic: -89.9672
  p-value: 0.0000
  Significant: Yes

JOB vs TARGET:
Conversion Rates:
  student: 31.43%
  retired: 25.23%
  unemployed: 14.20%
  admin.: 12.97%
  management: 11.22%
  unknown: 11.21%
  technician: 10.83%
  self-employed: 10.49%
  housemaid: 10.00%
  entrepreneur: 8.52%
  services: 8.14%
  blue-collar: 6.89%

Chi-square Test:
  Chi-square: 961.2424
  p-value: 0.0000
  Significant: Yes

EXECUTIVE SUMMARY
=================
Dataset: 41188 records x 21 variables
Target Distribution: {'no': 88.73458288821988, 'yes': 11.265417111780131}
Data Quality: Good

Strong Correlations Found: 8
  pdays ↔ previous: -0.588
  previous ↔ nr.employed: -0.501
  emp.var.rate ↔ cons.price.idx: 0.775
```

```
  emp.var.rate ↔ euribor3m: 0.972
  emp.var.rate ↔ nr.employed: 0.907
  cons.price.idx ↔ euribor3m: 0.688
  cons.price.idx ↔ nr.employed: 0.522
  euribor3m ↔ nr.employed: 0.945

Limitations:
```

In [19]:
```
# Generate the profile report
#profile = ProfileReport(df, title="My Data Profile Report")
# Display the report (e.g., in a Jupyter Notebook)
#profile.to_file("data_profile_report.html")
```

Fase 3: Preparación de los Datos (Data Preparation) 🛠️

In [20]:
```
df=remove_duplicates(df)
```

```
🗑️ 3.2 ELIMINACIÓN DE FILAS DUPLICADAS
--------------------------------------------------
Filas duplicadas encontradas: 12
✅ 12 filas duplicadas eliminadas
✅ Forma del dataset después: (41176, 21)
✅ Verificación final: 0 duplicados restantes
```

In [21]:
```
test_size=0.2
random_state=42
```

In [22]:
```
df.duplicated().sum()
```

Out[22]:
```
np.int64(0)
```

In [23]:
```
# FASE 3: Data Preparation
X, y, categorical_vars, numerical_vars = fase_3_data_preparation(df)
```

```
============================================================
FASE 3: DATA PREPARATION
============================================================

📋 3.1 ELIMINACIÓN DE VARIABLES PROBLEMÁTICAS
--------------------------------------------------
✅ Variable 'duration' eliminada (no disponible antes de llamada)

🔧 3.2 TRATAMIENTO DE VALORES FALTANTES
--------------------------------------------------
Variables con 'unknown': {'job': np.int64(330), 'marital': np.int64(80), 'education': np.int64(1730), 'default': np.int64(8596), 'housing': np.int64(990), 'loan': np.int64(990)}

⚙️ 3.3 FEATURE ENGINEERING
--------------------------------------------------

🔗 3.5 MANEJO DE MULTICOLINEALIDAD
--------------------------------------------------
✅ Variables económicas redundantes eliminadas: ['cons.price.idx', 'euribor3m', 'emp.var.rate', 'cons.conf.idx']
✅ Mantenida 'nr.employed' como representante del contexto económico

📊 3.6 SELECCIÓN FINAL DE CARACTERÍSTICAS
--------------------------------------------------
Variables categóricas finales (14): ['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact', 'month', 'day_of_week', 'poutcome', 'age_group', 'campaign_intensity', 'contact_history', 'economic_co
ntext']
Variables numéricas finales (5): ['age', 'campaign', 'pdays', 'previous', 'nr.employed']

✅ DATASET PREPARADO:
   • Forma final: (41176, 19)
   • Variables categóricas: 14
   • Variables numéricas: 5
   • Balance de clases: y
0    36537
1     4639
Name: count, dtype: int64
   • Porcentaje clase minoritaria: 11.27%
```

In [24]:
```
X
```

Out[24]:

	job	marital	education	default	housing	loan	contact	month	day_of_week	poutcome	age_group	campaign_intensity	contact_history	economic_context	age	campaign	pdays	previous	nr.emplo
0	housemaid	married	basic.4y	no	no	no	telephone	may	mon	nonexistent	senior	low	first_contact	high_employment	56	1	999	0	519
1	services	married	high.school	unknown	no	no	telephone	may	mon	nonexistent	senior	low	first_contact	high_employment	57	1	999	0	519
2	services	married	high.school	no	yes	no	telephone	may	mon	nonexistent	middle_young	low	first_contact	high_employment	37	1	999	0	519
3	admin.	married	basic.6y	no	no	no	telephone	may	mon	nonexistent	middle_young	low	first_contact	high_employment	40	1	999	0	519
4	services	married	high.school	no	no	yes	telephone	may	mon	nonexistent	senior	low	first_contact	high_employment	56	1	999	0	519
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
41183	retired	married	professional.course	no	yes	no	cellular	nov	fri	nonexistent	elderly	low	first_contact	low_employment	73	1	999	0	496
41184	blue-collar	married	professional.course	no	no	no	cellular	nov	fri	nonexistent	middle_old	low	first_contact	low_employment	46	1	999	0	496
41185	retired	married	university.degree	no	yes	no	cellular	nov	fri	nonexistent	senior	medium	first_contact	low_employment	56	2	999	0	496
41186	technician	married	professional.course	no	no	no	cellular	nov	fri	nonexistent	middle_old	low	first_contact	low_employment	44	1	999	0	496
41187	retired	married	professional.course	no	yes	no	cellular	nov	fri	failure	elderly	medium	previous_failure	low_employment	74	3	999	1	496

41176 rows × 19 columns

In [25]:
```
# Crear pipeline de preprocessing
preprocessor = crear_preprocessing_pipeline(categorical_vars, numerical_vars)
```

```
🔄 CREANDO PIPELINE DE PREPROCESSING
--------------------------------------------------
✅ Pipeline de preprocessing creado:
   • Variables numéricas: imputación mediana + estandarización
   • Variables categóricas: imputación 'unknown' + one-hot encoding
```

In [26]:
```
# División de datos
X_train, X_test, y_train, y_test = split_data_estratificado(X, y, test_size, random_state)
```

```
📂 DIVISIÓN ESTRATIFICADA DE DATOS
--------------------------------------------------
✅ División completada:
   • Train: 32940 muestras (80%)
   • Test: 8236 muestras (20%)
   • Distribución train: [88.73406193 11.26593807]
   • Distribución test: [88.73239437 11.26760563]
```

In [27]:
```
# División de datos
X_train, X_test, y_train, y_test = split_data_estratificado(X, y, test_size, random_state)
```

```
📂 DIVISIÓN ESTRATIFICADA DE DATOS
--------------------------------------------------
✅ División completada:
   • Train: 32940 muestras (80%)
   • Test: 8236 muestras (20%)
   • Distribución train: [88.73406193 11.26593807]
   • Distribución test: [88.73239437 11.26760563]
```

Fase 4: Modelado (Modeling) 🤖

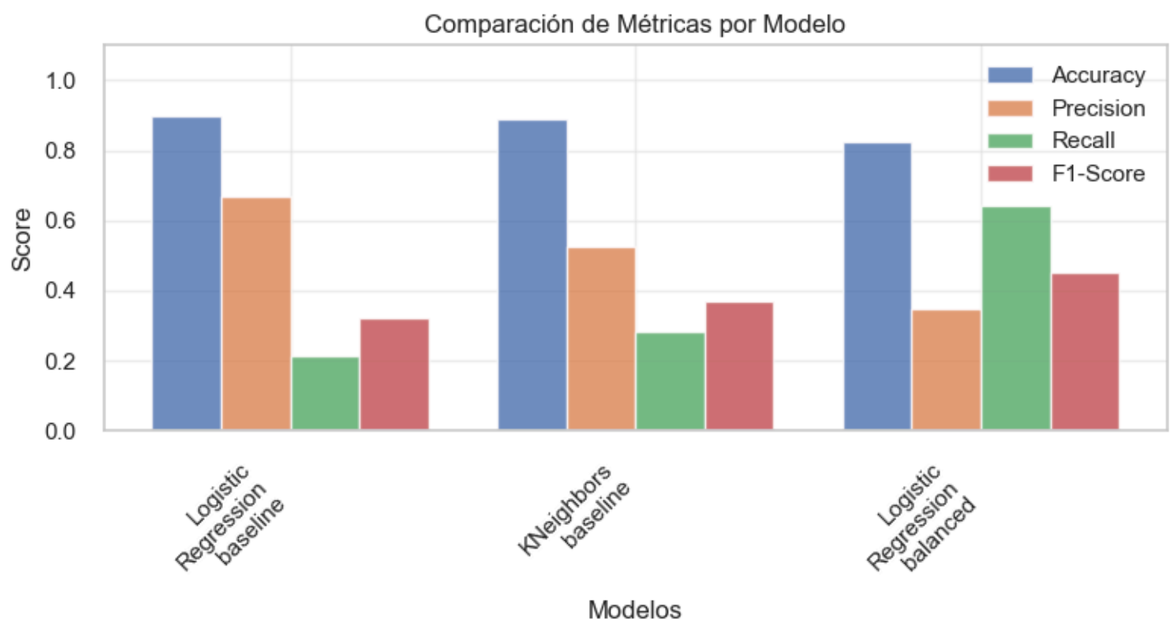
In [28]:
```
# FASE 4: Modeling
resultados_modelos = fase_4_modeling(X_train, X_test, y_train, y_test, preprocessor)
```

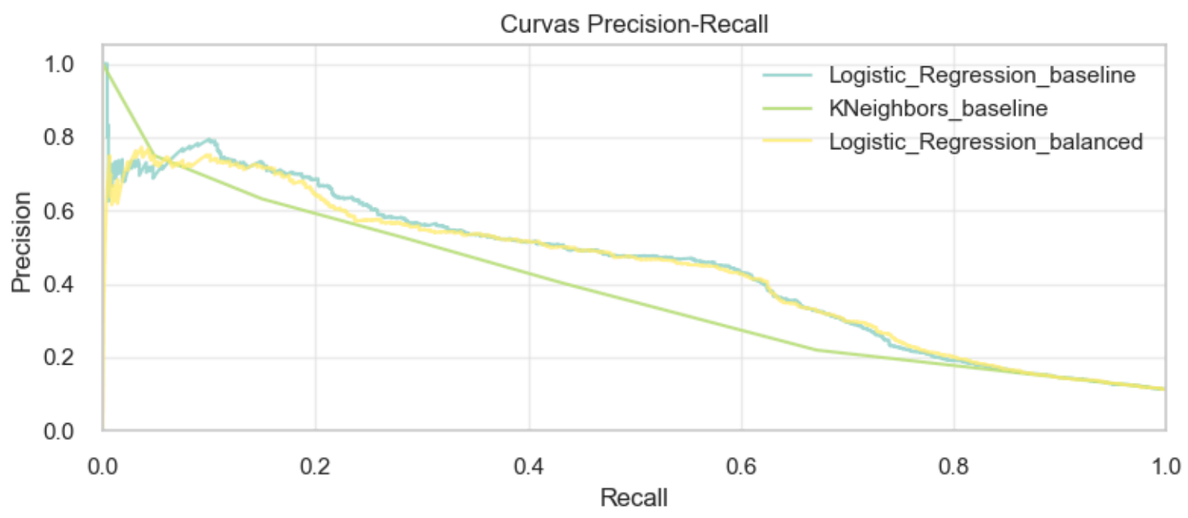
```
============================================================
FASE 4: MODELING
============================================================

👾 4.1 MODELOS BASELINE (SIN BALANCEO)
--------------------------------------------------

🔸 Entrenando Logistic_Regression...

🔸 Entrenando KNeighbors...

⚖️ 4.2 LOGISTIC REGRESSION CON CLASS_WEIGHT BALANCED
--------------------------------------------------

🔸 Entrenando Logistic_Regression + class_weight='balanced'...

📊 4.3 RESUMEN DE RESULTADOS
--------------------------------------------------

Comparación de modelos por F1-Score:
  Logistic_Regression_baseline: F1-Score = 0.3233, Accuracy = 0.8993
  KNeighbors_baseline: F1-Score = 0.3690, Accuracy = 0.8904
  Logistic_Regression_balanced: F1-Score = 0.4524, Accuracy = 0.8248

✅ Mejor modelo: Logistic_Regression_balanced
   F1-Score: 0.4524
   Accuracy: 0.8248
```

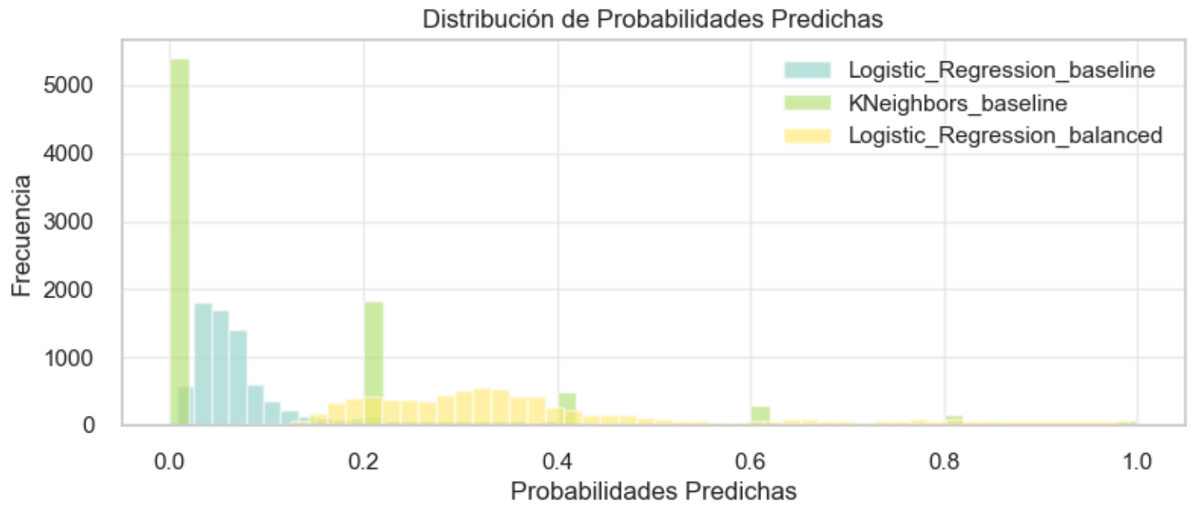
Fase 5: Evaluación (Evaluation) ✅

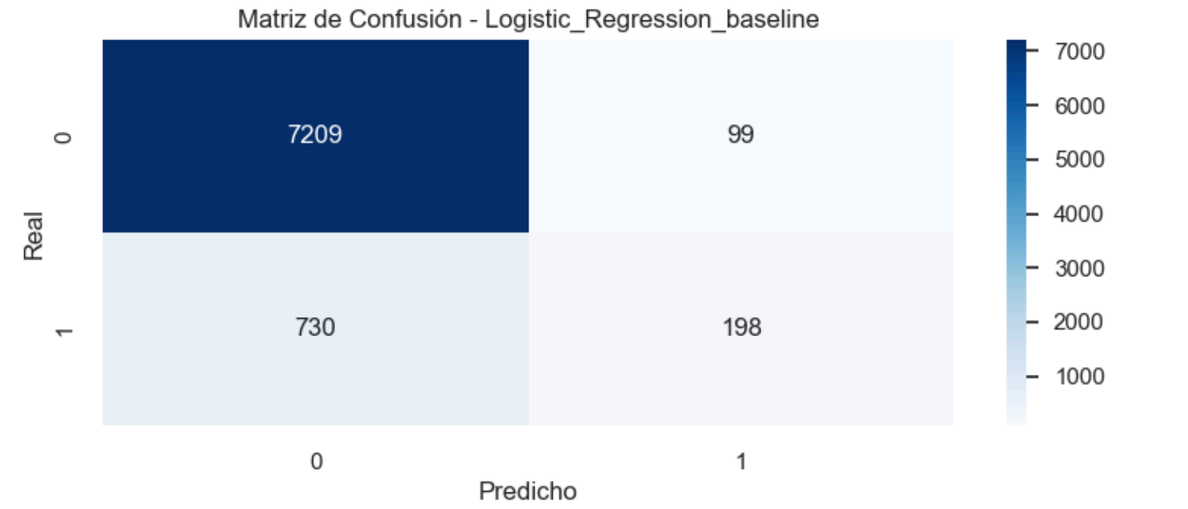
In [29]:
```
# Reporte final
df_comparativo, mejor_modelo = generar_reporte_final_modelos(resultados_modelos, y_test)
```

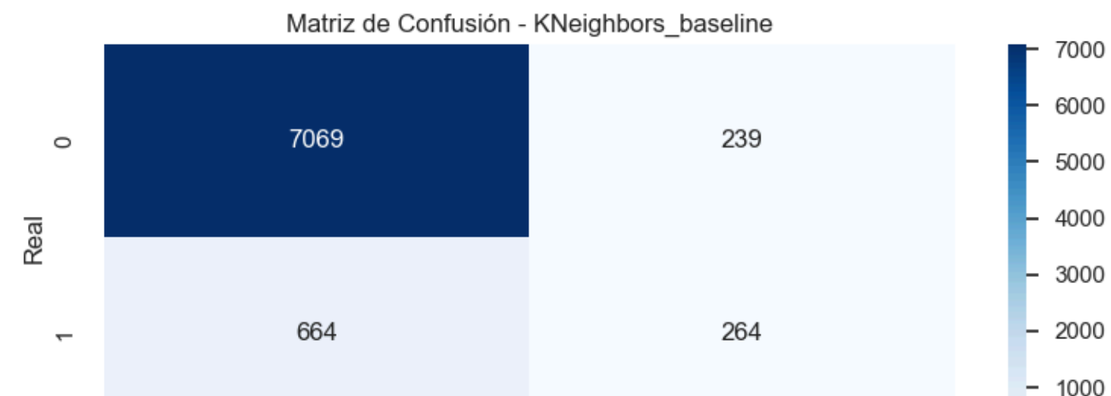
```
============================================================
REPORTE FINAL DE MODELOS
============================================================

📊 COMPARATIVA DE MODELOS:
                     Modelo Accuracy Precision Recall F1-Score
Logistic_Regression_balanced   0.8248    0.3492 0.6422   0.4524
                  KNeighbors   0.8904    0.5249 0.2845   0.3690
         Logistic_Regression   0.8993    0.6667 0.2134   0.3233

🏆 MEJOR MODELO: Logistic_Regression_balanced

📈 Generando visualizaciones finales...
```

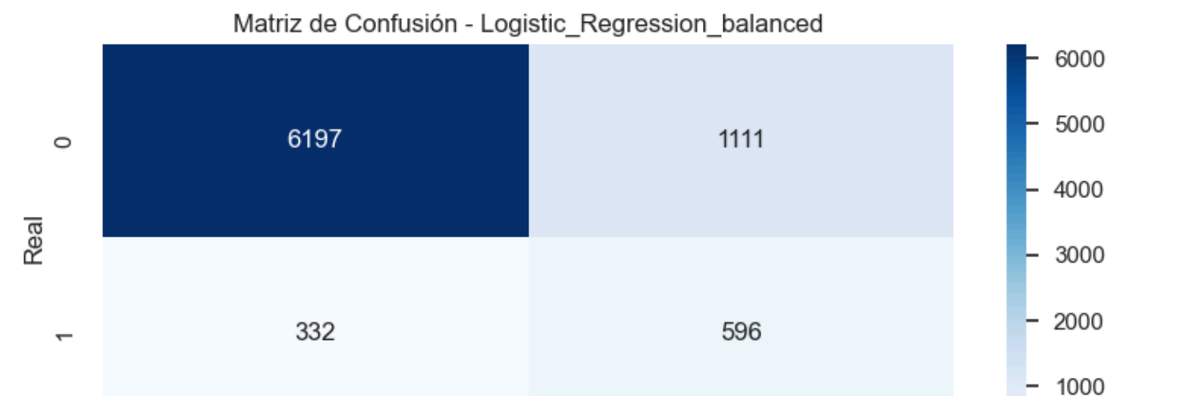

Comparación de Métricas por Modelo
Accuracy
Precision
Recall
F1-Score
Score
Modelos
Logistic Regression baseline
KNeighbors baseline
Logistic Regression balanced


Curvas Precision-Recall
Logistic_Regression_baseline
KNeighbors_baseline
Logistic_Regression_balanced
Precision
Recall


Distribución de Probabilidades Predichas
Logistic_Regression_baseline
KNeighbors_baseline
Logistic_Regression_balanced
Frecuencia
Probabilidades Predichas


Matriz de Confusión - Logistic_Regression_baseline
7209
99
730
198
Real
Predicho


Matriz de Confusión - KNeighbors_baseline
7069
239
664
264
Real


Matriz de Confusión - Logistic_Regression_balanced
6197
1111
332
596
Real

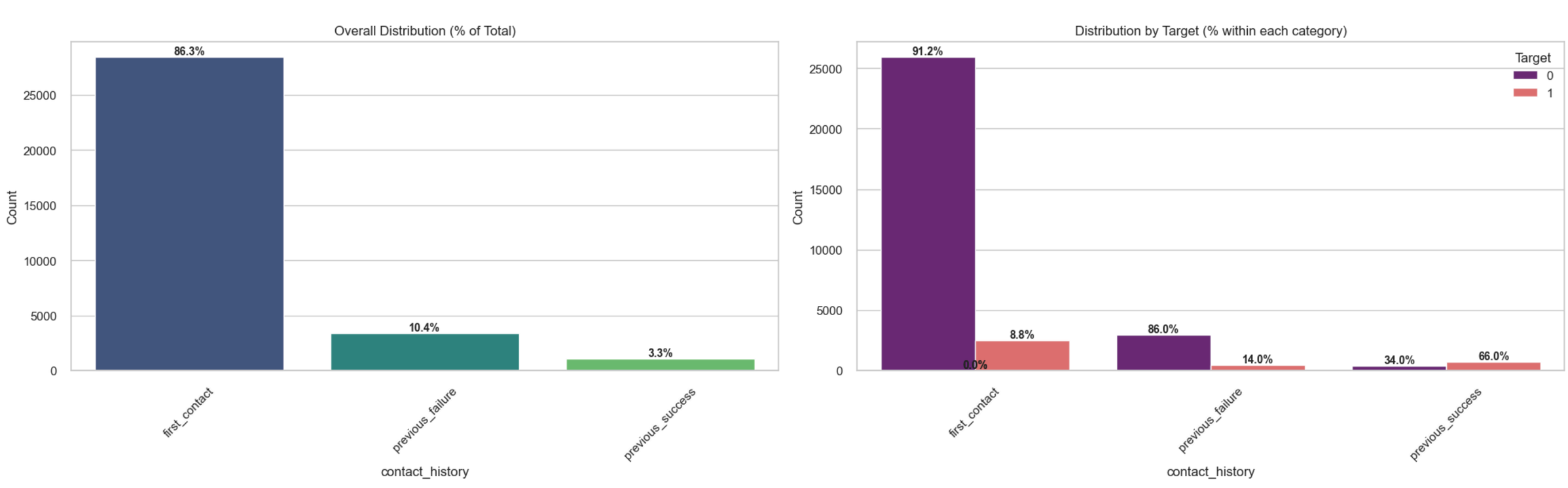
📊 RESUMEN DE MÉTRICAS
==

🔸 Logistic_Regression_baseline:
  Accuracy: 0.8993
  Precision: 0.6667
  Recall: 0.2134
  F1-Score: 0.3233

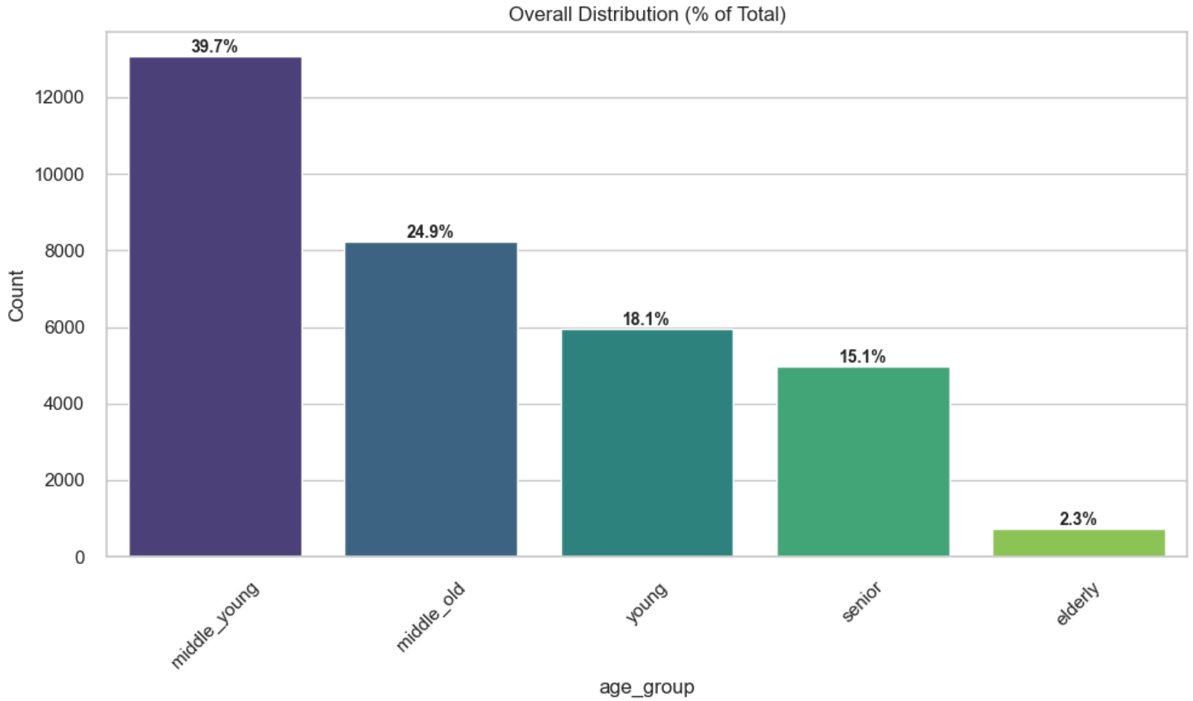
🔸 KNeighbors_baseline:
  Accuracy: 0.8904
  Precision: 0.5249
  Recall: 0.2845
  F1-Score: 0.3690

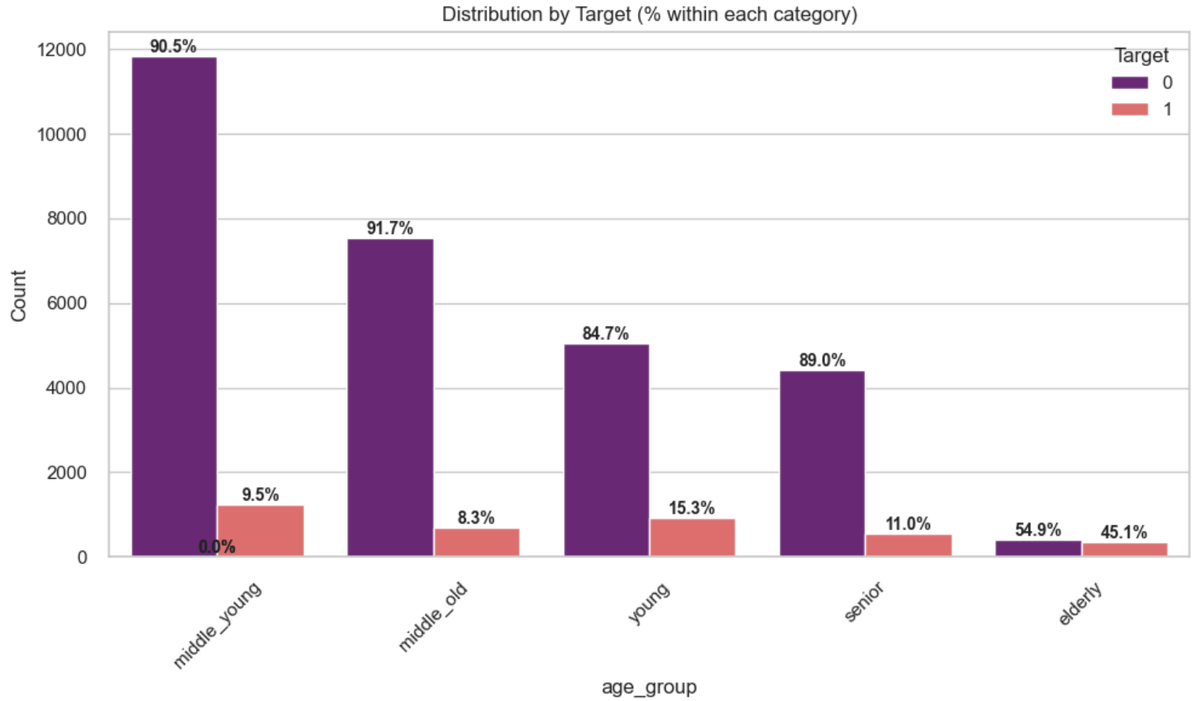
🔸 Logistic_Regression_balanced:
  Accuracy: 0.8248
  Precision: 0.3492
  Recall: 0.6422
  F1-Score: 0.4524

In [30]:
```
for current_category in ['contact_history', 'age_group','economic_context','contact_history','campaign_intensity']:
    plot_categorical_analysis_with_pct(X_train, y_train, current_category)
```

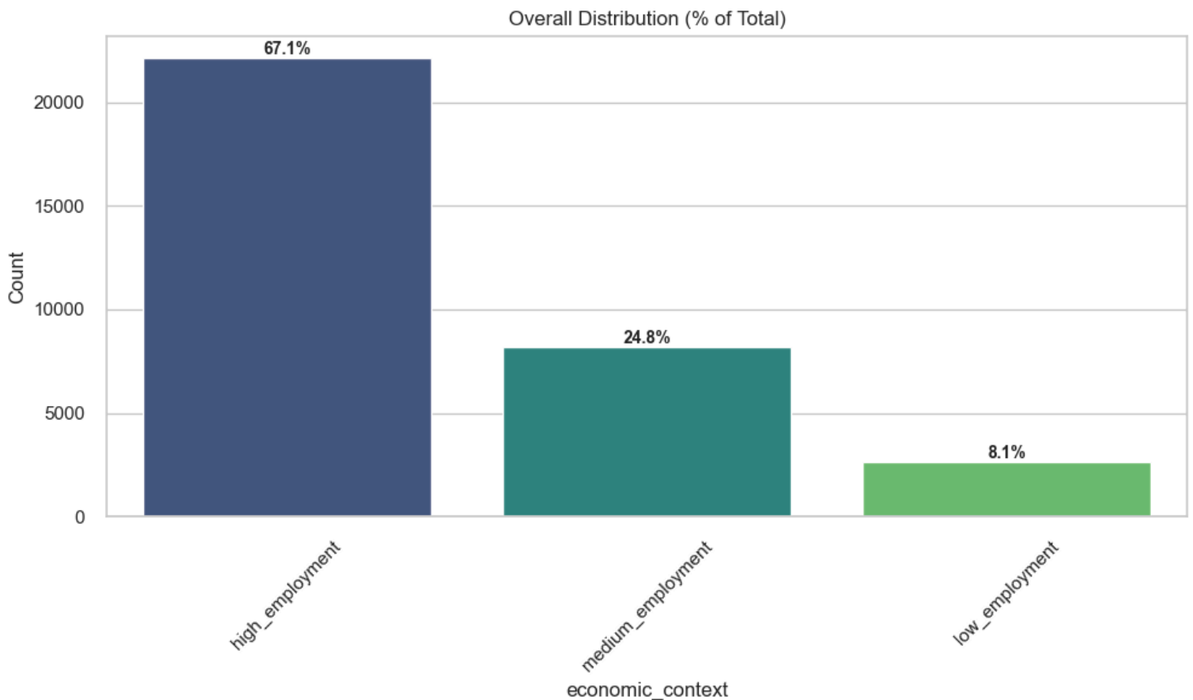


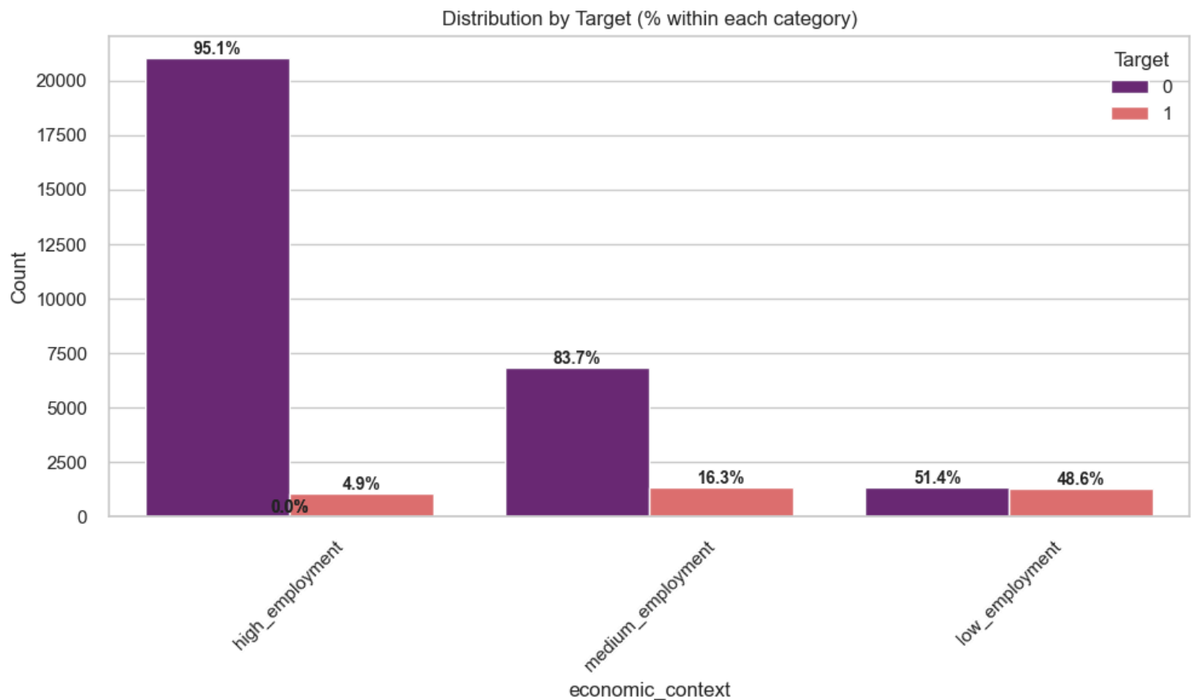
Analysis of Categorical Column: 'age_group'



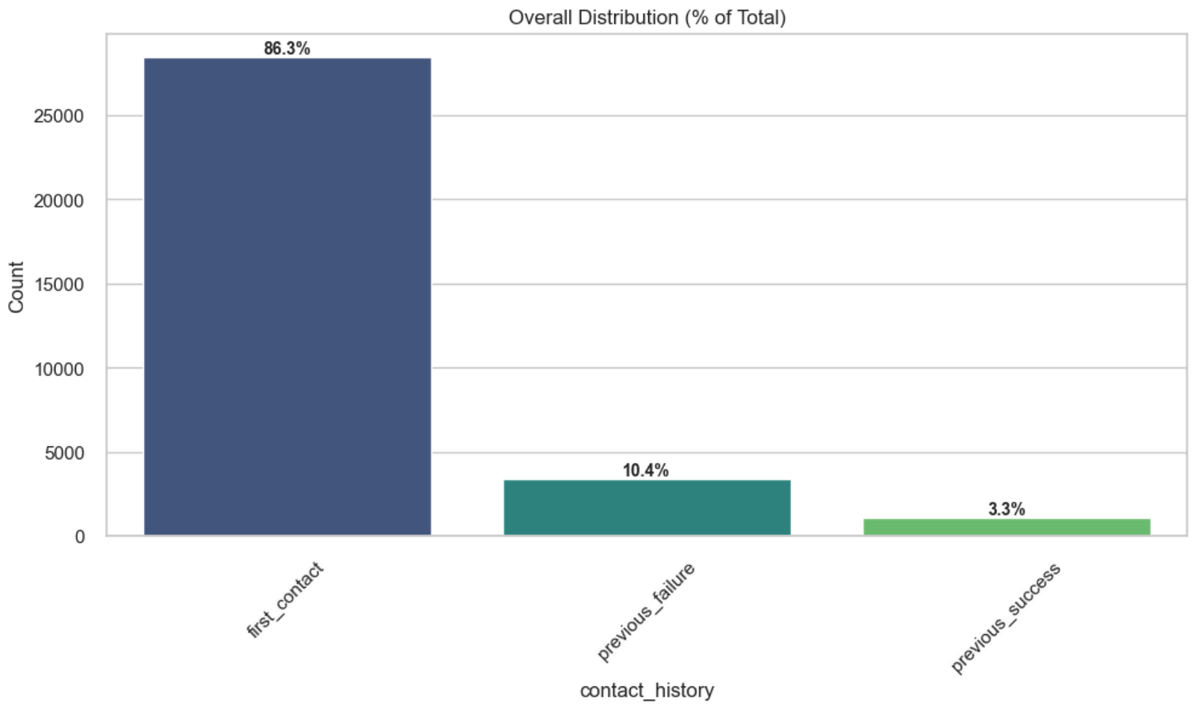


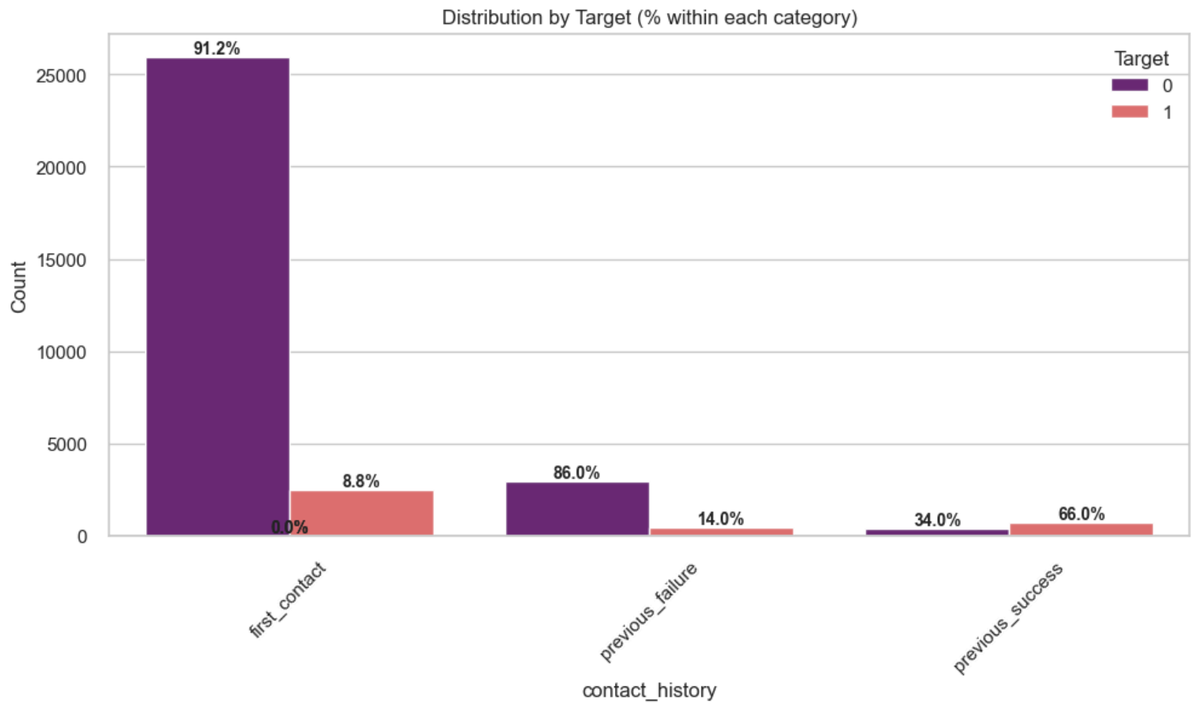
Analysis of Categorical Column: 'economic_context'



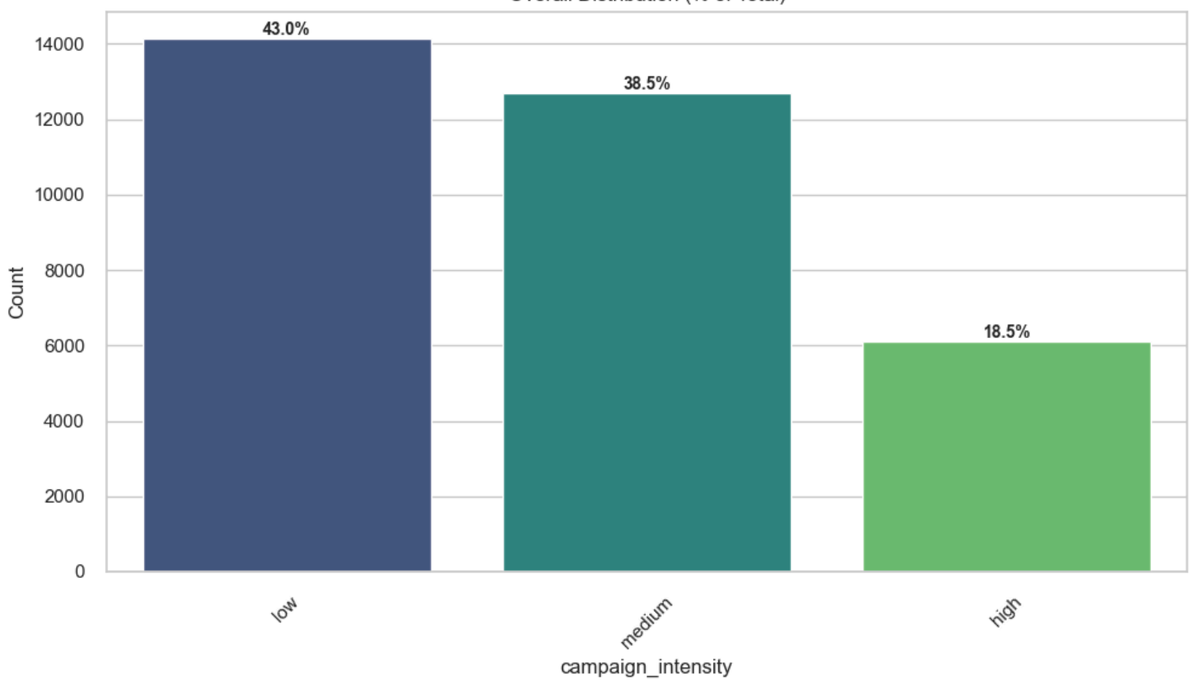


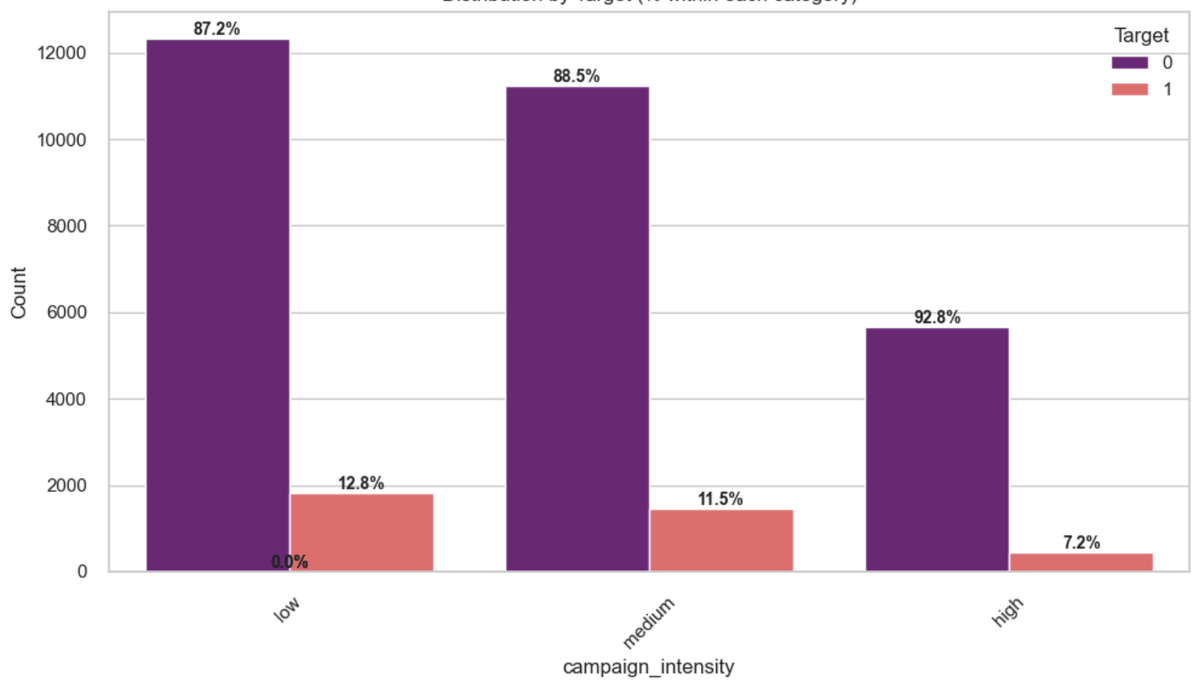
Analysis of Categorical Column: 'contact_history'





Analysis of Categorical Column: 'campaign_intensity'





Fase 6: Despliegue (Deployment) 🚀