

EDA

August 24, 2025

1 Importación de librerías

```
[1]: import numpy as np
import pandas as pd
import scipy.stats as ss
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

2 Importación del dataset

```
[2]: df = pd.read_csv('../Data/data_raw.csv')
```

3 Análisis exploratorio

3.1 Análisis general

```
[3]: df.info() # Información general del dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15420 entries, 0 to 15419
Data columns (total 33 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Month                 15420 non-null  object
1   WeekOfMonth           15420 non-null  int64
2   DayOfWeek             15420 non-null  object
3   Make                  15420 non-null  object
4   AccidentArea          15420 non-null  object
5   DayOfWeekClaimed      15420 non-null  object
6   MonthClaimed          15420 non-null  object
7   WeekOfMonthClaimed    15420 non-null  int64
8   Sex                   15420 non-null  object
9   MaritalStatus         15420 non-null  object
10  Age                   15420 non-null  int64
11  Fault                 15420 non-null  object
```

```

12 PolicyType          15420 non-null object
13 VehicleCategory     15420 non-null object
14 VehiclePrice        15420 non-null object
15 FraudFound_P        15420 non-null int64
16 PolicyNumber        15420 non-null int64
17 RepNumber           15420 non-null int64
18 Deductible          15420 non-null int64
19 DriverRating        15420 non-null int64
20 Days_Policy_Accident 15420 non-null object
21 Days_Policy_Claim    15420 non-null object
22 PastNumberOfClaims   15420 non-null object
23 AgeOfVehicle         15420 non-null object
24 AgeOfPolicyHolder    15420 non-null object
25 PoliceReportFiled    15420 non-null object
26 WitnessPresent      15420 non-null object
27 AgentType           15420 non-null object
28 NumberOfSuppliments  15420 non-null object
29 AddressChange_Claim  15420 non-null object
30 NumberOfCars        15420 non-null object
31 Year                15420 non-null int64
32 BasePolicy          15420 non-null object
dtypes: int64(9), object(24)
memory usage: 3.9+ MB

```

```
[4]: df.head() # Primer vistazo de los primeros 5 registros
```

```

[4]:   Month  WeekOfMonth  DayOfWeek  Make  AccidentArea  DayOfWeekClaimed  \
0    Dec             5  Wednesday  Honda           Urban           Tuesday
1    Jan             3  Wednesday  Honda           Urban           Monday
2    Oct             5    Friday  Honda           Urban           Thursday
3    Jun             2  Saturday  Toyota          Rural           Friday
4    Jan             5    Monday  Honda           Urban           Tuesday

   MonthClaimed  WeekOfMonthClaimed  Sex  MaritalStatus  ...  AgeOfVehicle  \
0          Jan                   1  Female           Single  ...      3 years
1          Jan                   4   Male           Single  ...      6 years
2          Nov                   2   Male           Married  ...      7 years
3          Jul                   1   Male           Married  ...  more than 7
4          Feb                   2  Female           Single  ...      5 years

   AgeOfPolicyHolder  PoliceReportFiled  WitnessPresent  AgentType  \
0          26 to 30                No                No  External
1          31 to 35                Yes                No  External
2          41 to 50                No                No  External
3          51 to 65                Yes                No  External
4          31 to 35                No                No  External

```

	NumberOfSuppliments	AddressChange_Claim	NumberOfCars	Year	BasePolicy
0	none	1 year	3 to 4	1994	Liability
1	none	no change	1 vehicle	1994	Collision
2	none	no change	1 vehicle	1994	Collision
3	more than 5	no change	1 vehicle	1994	Liability
4	none	no change	1 vehicle	1994	Collision

[5 rows x 33 columns]

3.2 Descripción de las variables

Variable	Descripción
Month	Mes en el que ocurrió el accidente.
WeekOfMonth	Semana del mes en el que ocurrió el accidente.
DayOfWeek	Día de la semana en el que ocurrió el accidente.
Make	Fabricante del vehículo involucrado en el siniestro.
AccidentArea	Área donde ocurrió el accidente (urbana o rural).
DayOfWeekClaimed	Día de la semana en el que se procesó la reclamación.
MonthClaimed	Mes en el que se procesó la reclamación.
WeekOfMonthClaimed	Semana del mes en la que se procesó la reclamación.
Sex	Género del asegurado.
MaritalStatus	Estado civil del asegurado.
Age	Edad del asegurado.
Fault	Indica si el asegurado tuvo la culpa del accidente.
PolicyType	Tipo de póliza de seguro.
VehicleCategory	Categoría del vehículo.
VehiclePrice	Precio del vehículo.
FraudFound_P	Variable objetivo: indica si se detectó fraude en la reclamación.
PolicyNumber	Identificador único de la póliza de seguro.
RepNumber	Identificador único del representante que gestionó la reclamación.
Deductible	Deducible que debe pagar el asegurado antes de que la aseguradora cubra los costos restantes.
DriverRating	Puntuación del conductor.
Days_Policy_Accident	Días desde la emisión de la póliza hasta el accidente.
Days_Policy_Claim	Días desde la emisión de la póliza hasta la reclamación.
PastNumberOfClaims	Número de reclamaciones anteriores del asegurado.
AgeOfVehicle	Antigüedad del vehículo.
AgeOfPolicyHolder	Edad del asegurado.
PoliceReportFiled	Indica si se presentó un informe policial.
WitnessPresent	Indica si hubo testigos presentes en el accidente.
AgentType	Tipo de agente que gestionó la póliza (interno o externo).
NumberOfSuppliments	Número de documentos o reclamaciones suplementarias relacionadas con el caso.
AddressChange_Claim	Indica si el asegurado cambió de domicilio en el momento de la reclamación.
NumberOfCars	Número de vehículos involucrados en el accidente.
Year	Año en el que se realizó o procesó la reclamación.

Variable	Descripción
BasePolicy	Tipo de póliza base.

3.3 Valores únicos

[5]: *# Tabla para visualizar el número de valores únicos de cada variable y cada uno de esos valores*

```
resumen_unicos = []

for col in df.columns:
    valores = df[col].value_counts(dropna=False).sort_index()
    resumen_unicos.append({
        "Variable": col,
        "Núm. Valores únicos": df[col].nunique(),
        "Valores": valores.to_dict()
    })

resumen_df = pd.DataFrame(resumen_unicos)
resumen_df
```

```
[5]:
```

	Variable	Núm. Valores únicos	\
0	Month	12	
1	WeekOfMonth	5	
2	DayOfWeek	7	
3	Make	19	
4	AccidentArea	2	
5	DayOfWeekClaimed	8	
6	MonthClaimed	13	
7	WeekOfMonthClaimed	5	
8	Sex	2	
9	MaritalStatus	4	
10	Age	66	
11	Fault	2	
12	PolicyType	9	
13	VehicleCategory	3	
14	VehiclePrice	6	
15	FraudFound_P	2	
16	PolicyNumber	15420	
17	RepNumber	16	
18	Deductible	4	
19	DriverRating	4	
20	Days_Policy_Accident	5	
21	Days_Policy_Claim	4	
22	PastNumberOfClaims	4	
23	AgeOfVehicle	8	

24	AgeOfPolicyHolder	9
25	PoliceReportFiled	2
26	WitnessPresent	2
27	AgentType	2
28	NumberOfSuppliments	4
29	AddressChange_Claim	5
30	NumberOfCars	5
31	Year	3
32	BasePolicy	3

Valores

```

0  {'Apr': 1280, 'Aug': 1127, 'Dec': 1285, 'Feb':...
1    {1: 3187, 2: 3558, 3: 3640, 4: 3398, 5: 1637}
2  {'Friday': 2445, 'Monday': 2616, 'Saturday': 1...
3  {'Accura': 472, 'BMW': 15, 'Chevrolet': 1681, ...
4    {'Rural': 1598, 'Urban': 13822}
5  {'0': 1, 'Friday': 2497, 'Monday': 3757, 'Satu...
6  {'0': 1, 'Apr': 1271, 'Aug': 1126, 'Dec': 1146...
7    {1: 3450, 2: 3720, 3: 3583, 4: 3433, 5: 1234}
8    {'Female': 2420, 'Male': 13000}
9  {'Divorced': 76, 'Married': 10625, 'Single': 4...
10 {0: 320, 16: 9, 17: 6, 18: 48, 19: 32, 20: 28,...
11   {'Policy Holder': 11230, 'Third Party': 4190}
12 {'Sedan - All Perils': 4087, 'Sedan - Collisio...
13   {'Sedan': 9671, 'Sport': 5358, 'Utility': 391}
14 {'20000 to 29000': 8079, '30000 to 39000': 353...
15   {0: 14497, 1: 923}
16 {1: 1, 2: 1, 3: 1, 4: 1, 5: 1, 6: 1, 7: 1, 8: ...
17 {1: 987, 2: 956, 3: 949, 4: 912, 5: 987, 6: 94...
18   {300: 8, 400: 14838, 500: 263, 700: 311}
19   {1: 3944, 2: 3801, 3: 3884, 4: 3791}
20 {'1 to 7': 14, '15 to 30': 49, '8 to 15': 55, ...
21 {'15 to 30': 56, '8 to 15': 21, 'more than 30'...
22 {'1': 3573, '2 to 4': 5485, 'more than 4': 201...
23 {'2 years': 73, '3 years': 152, '4 years': 229...
24 {'16 to 17': 320, '18 to 20': 15, '21 to 25': ...
25   {'No': 14992, 'Yes': 428}
26   {'No': 15333, 'Yes': 87}
27   {'External': 15179, 'Internal': 241}
28 {'1 to 2': 2489, '3 to 5': 2017, 'more than 5'...
29 {'1 year': 170, '2 to 3 years': 291, '4 to 8 y...
30 {'1 vehicle': 14316, '2 vehicles': 709, '3 to ...
31   {1994: 6142, 1995: 5195, 1996: 4083}
32 {'All Perils': 4449, 'Collision': 5962, 'Liabi...
```

3.3.1 Primeras observaciones

Variables categóricas simples

- **AccidentArea** → Convertir a variable booleana (**Urban**=1, **Rural**=0).
- **Sex** → Convertir a booleana (**Male**=1, **Female**=0).
- **Fault** → Convertir a booleana (**Third Party**=1, **Policy Holder**=0).
- **PoliceReportFiled** → Convertir a booleana (**Yes**=1, **No**=0).
- **WitnessPresent** → Convertir a booleana (**Yes**=1, **No**=0).
- **AgentType** → Convertir a booleana (**Internal**=1, **External**=0).

Variables con valores erróneos

- **DayOfWeekClaimed** → 8 valores únicos; el valor 0 es erróneo → eliminar (solo 1 caso).
- **MonthClaimed** → 13 valores únicos; el valor 0 es erróneo → eliminar (solo 1 caso).
- **Age** → Contiene 0 (no válido como edad) → imputar (hay bastantes casos).

Variables redundantes o poco informativas

- **PolicyType** → Parece la combinación de **VehicleCategory** y **BasePolicy**. A analizar su redundancia.
- **Age** y **AgeOfPolicyHolder** → Parecen la misma variable. Como **Age** tiene muchos valores 0, se opta por eliminar esta variable y mantener la segunda.
- **PolicyNumber** → Identificador único, sin valor predictivo.
- **RepNumber** → Identificador único, sin valor predictivo.
- **Year** → Variable temporal que no puede usarse en un entorno real porque no generaliza a años futuro.

Variables con valores en rangos que requieren conversión Para estas variables categóricas con intervalos de valores, reemplazar cada rango por el valor medio y convertir a **float**: - **VehiclePrice** - **Days_Policy_Accident** - **Days_Policy_Claim** - **AgeOfVehicle** - **NumberOfCars** - **PastNumberOfClaims** - **AgeOfPolicyHolder** - **NumberOfSupplements** - **AddressChange_Claim**

Variables que no tienen clarificación

- **DriverRating** → ¿En qué se basa esa puntuación? Eliminar por falta de información.

Variables categóricas a codificar A continuación, se identifican las variables categóricas que necesitan ser transformadas antes de ser utilizadas en el modelo: - **Month** - **DayOfWeek** - **Make** - **DayOfWeekClaimed** - **MonthClaimed** - **MaritalStatus** - **VehicleCategory** - **BasePolicy** - **AddressChange_Claim**

Problemas de desbalance

- **FraudFound_P** → Variable objetivo muy desbalanceada (~6% casos positivos). Se valora aplicar SMOTE en el modelo.

3.4 Valores atípicos

```
[6]: # Para variables numéricas:
    ## Cálculo del coeficiente de asimetría (skewness)

df.select_dtypes(include=np.number).skew()
```

```
[6]: WeekOfMonth          0.115426
     WeekOfMonthClaimed    0.158233
     Age                  0.152314
     FraudFound_P         3.711164
     PolicyNumber          0.000000
     RepNumber             0.006628
     Deductible            6.078803
     DriverRating          0.009283
     Year                  0.245689
     dtype: float64
```

```
[7]: # Para variables numéricas:
    ## Extracción de estadísticas clave: media, desviación estándar (std), mínimo,
    ↪ máximo, y los percentiles 25%, 50% y 75%

df.describe(include=np.number)
```

```
[7]:
```

	WeekOfMonth	WeekOfMonthClaimed	Age	FraudFound_P	\
count	15420.000000	15420.000000	15420.000000	15420.000000	
mean	2.788586	2.693969	39.855707	0.059857	
std	1.287585	1.259115	13.492377	0.237230	
min	1.000000	1.000000	0.000000	0.000000	
25%	2.000000	2.000000	31.000000	0.000000	
50%	3.000000	3.000000	38.000000	0.000000	
75%	4.000000	4.000000	48.000000	0.000000	
max	5.000000	5.000000	80.000000	1.000000	

	PolicyNumber	RepNumber	Deductible	DriverRating	Year
count	15420.000000	15420.000000	15420.000000	15420.000000	15420.000000
mean	7710.500000	8.483268	407.704280	2.487808	1994.866472
std	4451.514911	4.599948	43.950998	1.119453	0.803313
min	1.000000	1.000000	300.000000	1.000000	1994.000000
25%	3855.750000	5.000000	400.000000	1.000000	1994.000000
50%	7710.500000	8.000000	400.000000	2.000000	1995.000000
75%	11565.250000	12.000000	400.000000	3.000000	1996.000000
max	15420.000000	16.000000	700.000000	4.000000	1996.000000

```
[8]: # Para variables categóricas:
## Extracción de estadísticas clave: top y frecuencia

df.describe(exclude=np.number)
```

```
[8]:      Month DayOfWeek      Make AccidentArea DayOfWeekClaimed MonthClaimed \
count    15420      15420      15420      15420      15420      15420
unique      12         7         19         2         8         13
top        Jan    Monday  Pontiac      Urban      Monday      Jan
freq      1411      2616      3837      13822      3757      1446

      Sex MaritalStatus      Fault      PolicyType ... \
count    15420      15420      15420      15420 ...
unique      2         4         2         9 ...
top      Male    Married  Policy Holder  Sedan - Collision ...
freq    13000      10625      11230      5584 ...

      PastNumberOfClaims AgeOfVehicle AgeOfPolicyHolder PoliceReportFiled \
count              15420      15420      15420      15420
unique              4         8         9         2
top              2 to 4    7 years    31 to 35      No
freq              5485      5807      5593      14992

      WitnessPresent AgentType NumberOfSupplements AddressChange_Claim \
count              15420      15420      15420      15420
unique              2         2         4         5
top              No  External      none      no change
freq            15333      15179      7047      14324

      NumberOfCars BasePolicy
count          15420      15420
unique           5         3
top      1 vehicle  Collision
freq          14316      5962

[4 rows x 24 columns]
```

3.4.1 Observaciones sobre las variables numéricas

1. Age

- Mínimo = 0 → valor inválido.
- Máximo = 80 → razonable.
- Media 39.86 → predominan adultos de mediana edad.

2. FraudFound_P

- Media 0.059 → confirma fuerte desbalance (~6% casos positivos).

3. PolicyNumber

- Rango 0–15420 → identificador único sin valor predictivo. Se eliminará.
4. **RepNumber**
 - Rango 1–16 → código interno del representante. Se eliminará.
 5. **Deductible**
 - Valores entre 300 y 700 esperados, pero predominio claro en 400.
 6. **DriverRating**
 - Rango 1–4 → variable ordinal. Se eliminará.
 7. **Year**
 - Rango 1994–1996 → dataset solo cubre 3 años.
 8. **WeekOfMonth / WeekOfMonthClaimed**
 - Rango 1–5.
- Medias ~2.7

3.5 Valores faltantes

```
[9]: # Recuento de valores NaN explícitos
```

```
df.isnull().sum()
```

```
[9]: Month                0
WeekOfMonth              0
DayOfWeek                0
Make                    0
AccidentArea             0
DayOfWeekClaimed         0
MonthClaimed             0
WeekOfMonthClaimed       0
Sex                      0
MaritalStatus            0
Age                      0
Fault                    0
PolicyType               0
VehicleCategory          0
VehiclePrice             0
FraudFound_P             0
PolicyNumber             0
RepNumber                0
Deductible               0
DriverRating             0
Days_Policy_Accident     0
Days_Policy_Claim        0
PastNumberOfClaims       0
AgeOfVehicle             0
AgeOfPolicyHolder        0
PoliceReportFiled        0
WitnessPresent           0
```

```

AgentType          0
NumberOfSupplements 0
AddressChange_Claim 0
NumberOfCars        0
Year                0
BasePolicy          0
dtype: int64

```

Se confirma que no existen valores *missing*.

3.6 Análisis de colinealidad y correlación

3.6.1 Análisis variables numéricas

```

[10]: # Mapa de calor de correlaciones (heatmap) con degradado de color para
      ↪ identificar visualmente relaciones destacadas entre pares de variables

```

```

corr = df.select_dtypes(include=np.number).corr()
corr.style.background_gradient(cmap='coolwarm').format(precision=3)

```

```

[10]: <pandas.io.formats.style.Styler at 0x1ba73b15710>

```

```

[11]: # Clasificación de correlaciones absolutas respecto a la variable objetivo,
      ↪ ordenadas de mayor a menor.

```

```

corr = abs(df.select_dtypes(include=np.number).corr())
corr[['FraudFound_P']].sort_values(by = 'FraudFound_P', ascending = False)

```

```

[11]:
FraudFound_P
FraudFound_P      1.000000
Age                0.029741
Year              0.024760
PolicyNumber      0.020345
Deductible        0.017348
WeekOfMonth       0.011861
RepNumber         0.007551
DriverRating      0.007266
WeekOfMonthClaimed 0.005761

```

Observaciones **Magnitudes muy bajas** - Ninguna variable numérica presenta correlación > 0.03 con **FraudFound_P**. - Esto confirma que no hay una variable numérica aislada que sea un predictor fuerte de fraude. - El poder predictivo probablemente esté en combinaciones de variables y patrones no lineales.

Otras observaciones: - El modelo deberá explotar interacciones y relaciones no lineales, lo que sugiere que algoritmos como árboles de decisión (XGBoost o LightGBM) pueden ser más adecuados que modelos lineales simples.

3.6.2 Análisis variables categóricas

```
[12]: # Función V de Cramér
def cramers_v(x, y):
    tab = pd.crosstab(x, y)
    chi2 = ss.chi2_contingency(tab)[0]
    n = tab.values.sum()
    phi2 = chi2 / n
    r, k = tab.shape
    # Corrección por sesgo
    phi2corr = max(0, phi2 - ((k-1)*(r-1))/(n-1))
    rcorr = r - ((r-1)**2)/(n-1)
    kcorr = k - ((k-1)**2)/(n-1)
    denom = min((kcorr-1), (rcorr-1))
    return np.sqrt(phi2corr / denom) if denom > 0 else np.nan

# Selección de categóricas
MAX_CARDINALITY = 50
cat_cols = [
    c for c in df.columns
    if (
        (df[c].dtype == 'object' or str(df[c].dtype) == 'category')
        or (pd.api.types.is_integer_dtype(df[c]) and df[c].nunique() <= 15)
    )
]

# Ajustes varios
target = 'FraudFound_P'
cat_cols = [c for c in cat_cols if c != target]

to_exclude = {'PolicyNumber', 'RepNumber', 'Year', 'DriverRating'}
cat_cols = [c for c in cat_cols if c not in to_exclude]

y = df[target].astype(str)

# Cálculo en bloque
rows = []
for c in cat_cols:
    x = df[c].astype(str) # tratar NaN como 'nan'
    # (opcional) saltar columnas con cardinalidad desmesurada
    if x.nunique(dropna=False) > MAX_CARDINALITY:
        rows.append({"variable": c, "cardinalidad": x.nunique(), "v_cramer": np.
        nan})
        continue
    v = cramers_v(x, y)
    rows.append({"variable": c, "cardinalidad": x.nunique(), "v_cramer": v})
```

```

cramer_df = (
    pd.DataFrame(rows)
    .sort_values("v_cramer", ascending=False)
    .reset_index(drop=True)
)

cramer_df

```

```

[12]:
      variable  cardinalidad  v_cramer
0      PolicyType           9  0.166880
1      BasePolicy           3  0.161237
2  VehicleCategory           3  0.136892
3          Fault           2  0.130839
4  AddressChange_Claim       5  0.080827
5          Deductible        4  0.067096
6      VehiclePrice           6  0.063803
7  PastNumberOfClaims         4  0.057230
8          Make            19  0.052072
9      MonthClaimed          13  0.044305
10  AgeOfPolicyHolder         9  0.040269
11          Month           12  0.034914
12      AccidentArea          2  0.032056
13  NumberOfSupplements        4  0.031336
14      AgeOfVehicle           8  0.031116
15          Sex             2  0.028461
16  Days_Policy_Accident        5  0.022159
17          AgentType         2  0.020341
18      DayOfWeek             7  0.016406
19  PoliceReportFiled          2  0.012863
20  Days_Policy_Claim          4  0.011045
21      WeekOfMonth           5  0.000000
22  DayOfWeekClaimed           8  0.000000
23      MaritalStatus          4  0.000000
24      WitnessPresent         2  0.000000
25      NumberOfCars           5  0.000000
26  WeekOfMonthClaimed         5  0.000000

```

```

[13]: # Matriz vacía
redundancy_matrix = pd.DataFrame(index=cat_cols, columns=cat_cols, dtype=float)

# Cálculo
for col1 in cat_cols:
    for col2 in cat_cols:
        if col1 == col2:
            redundancy_matrix.loc[col1, col2] = 1.0
        else:
            redundancy_matrix.loc[col1, col2] = cramers_v(

```

```

        df[col1].astype(str),
        df[col2].astype(str)
    )

redundancy_matrix

```

```

[13]:
      Month  WeekOfMonth  DayOfWeek  Make \
Month      1.000000    0.041288    0.038230  0.000000
WeekOfMonth 0.041288    1.000000    0.015834  0.005024
DayOfWeek   0.038230    0.015834    1.000000  0.000000
Make        0.000000    0.005024    0.000000  1.000000
AccidentArea 0.011199    0.008196    0.024085  0.046078
DayOfWeekClaimed 0.033688    0.009069    0.143995  0.000000
MonthClaimed 0.747048    0.034555    0.011495  0.000000
WeekOfMonthClaimed 0.055586    0.401083    0.000000  0.000000
Sex          0.016662    0.000000    0.019061  0.073264
MaritalStatus 0.000000    0.010735    0.016736  0.067654
Fault        0.009280    0.019905    0.038686  0.046709
PolicyType    0.017798    0.023917    0.029341  0.165184
VehicleCategory 0.018287    0.017377    0.052441  0.192175
VehiclePrice  0.017490    0.000000    0.016892  0.260183
Deductible    0.006343    0.000000    0.017547  0.009295
Days_Policy_Accident 0.000000    0.014556    0.010988  0.018159
Days_Policy_Claim 0.009792    0.007332    0.005856  0.000000
PastNumberOfClaims 0.017109    0.004506    0.015877  0.033761
AgeOfVehicle   0.042221    0.000000    0.013079  0.136873
AgeOfPolicyHolder 0.031756    0.010117    0.021922  0.120186
PoliceReportFiled 0.048222    0.002525    0.012841  0.009755
WitnessPresent 0.000000    0.000000    0.000000  0.000000
AgentType      0.018577    0.000000    0.000000  0.033383
NumberOfSuppliments 0.015619    0.000000    0.005012  0.050781
AddressChange_Claim 0.007780    0.000000    0.000000  0.000000
NumberOfCars   0.059911    0.000000    0.000000  0.027050
BasePolicy     0.027061    0.006613    0.043162  0.114251

      AccidentArea  DayOfWeekClaimed  MonthClaimed \
Month      0.011199      0.033688      0.747048
WeekOfMonth 0.008196      0.009069      0.034555
DayOfWeek   0.024085      0.143995      0.011495
Make        0.046078      0.000000      0.000000
AccidentArea 1.000000      0.025763      0.028732
DayOfWeekClaimed 0.025763      1.000000      0.379825
MonthClaimed 0.028732      0.379825      1.000000
WeekOfMonthClaimed 0.008328      0.034784      0.062432
Sex          0.032531      0.000000      0.013365
MaritalStatus 0.000000      0.001638      0.000000
Fault        0.002637      0.018565      0.011617

```

PolicyType	0.067619	0.027996	0.017422
VehicleCategory	0.063997	0.020026	0.014175
VehiclePrice	0.018663	0.000000	0.024735
Deductible	0.000000	0.004408	0.000000
Days_Policy_Accident	0.000000	0.000000	0.000000
Days_Policy_Claim	0.020319	0.577352	0.577017
PastNumberOfClaims	0.061275	0.003849	0.021527
AgeOfVehicle	0.017491	0.019671	0.050035
AgeOfPolicyHolder	0.019578	0.023098	0.044539
PoliceReportFiled	0.000000	0.010614	0.058189
WitnessPresent	0.025711	0.009409	0.000000
AgentType	0.000000	0.022069	0.026430
NumberOfSuppliments	0.014681	0.000000	0.015780
AddressChange_Claim	0.025268	0.000000	0.000000
NumberOfCars	0.006159	0.003409	0.041025
BasePolicy	0.055858	0.015649	0.032798

	WeekOfMonthClaimed	Sex	MaritalStatus	...	\
Month	0.055586	0.016662	0.000000	...	
WeekOfMonth	0.401083	0.000000	0.010735	...	
DayOfWeek	0.000000	0.019061	0.016736	...	
Make	0.000000	0.073264	0.067654	...	
AccidentArea	0.008328	0.032531	0.000000	...	
DayOfWeekClaimed	0.034784	0.000000	0.001638	...	
MonthClaimed	0.062432	0.013365	0.000000	...	
WeekOfMonthClaimed	1.000000	0.000000	0.005955	...	
Sex	0.000000	1.000000	0.155753	...	
MaritalStatus	0.005955	0.155753	1.000000	...	
Fault	0.000000	0.000000	0.000000	...	
PolicyType	0.013266	0.092137	0.043917	...	
VehicleCategory	0.016659	0.082461	0.044580	...	
VehiclePrice	0.015760	0.146593	0.070636	...	
Deductible	0.000000	0.016676	0.018958	...	
Days_Policy_Accident	0.000000	0.000000	0.019803	...	
Days_Policy_Claim	0.006032	0.000000	0.009557	...	
PastNumberOfClaims	0.015204	0.000000	0.013760	...	
AgeOfVehicle	0.000000	0.212081	0.265808	...	
AgeOfPolicyHolder	0.000000	0.135169	0.304052	...	
PoliceReportFiled	0.023282	0.000000	0.000000	...	
WitnessPresent	0.000000	0.000000	0.012698	...	
AgentType	0.006986	0.008846	0.012194	...	
NumberOfSuppliments	0.009305	0.008177	0.020860	...	
AddressChange_Claim	0.006687	0.000000	0.000000	...	
NumberOfCars	0.000000	0.000000	0.012541	...	
BasePolicy	0.013382	0.068938	0.036605	...	

PastNumberOfClaims	AgeOfVehicle	AgeOfPolicyHolder	\
--------------------	--------------	-------------------	---

Month	0.017109	0.042221	0.031756
WeekOfMonth	0.004506	0.000000	0.010117
DayOfWeek	0.015877	0.013079	0.021922
Make	0.033761	0.136873	0.120186
AccidentArea	0.061275	0.017491	0.019578
DayOfWeekClaimed	0.003849	0.019671	0.023098
MonthClaimed	0.021527	0.050035	0.044539
WeekOfMonthClaimed	0.015204	0.000000	0.000000
Sex	0.000000	0.212081	0.135169
MaritalStatus	0.013760	0.265808	0.304052
Fault	0.126672	0.043507	0.055378
PolicyType	0.231660	0.083601	0.106387
VehicleCategory	0.237430	0.067292	0.075888
VehiclePrice	0.090318	0.192351	0.178653
Deductible	0.000000	0.077220	0.051775
Days_Policy_Accident	0.029486	0.034691	0.018085
Days_Policy_Claim	0.025142	0.034077	0.033518
PastNumberOfClaims	1.000000	0.036825	0.031441
AgeOfVehicle	0.036825	1.000000	0.534242
AgeOfPolicyHolder	0.031441	0.534242	1.000000
PoliceReportFiled	0.002223	0.004303	0.000000
WitnessPresent	0.016320	0.027724	0.013846
AgentType	0.023476	0.013653	0.000000
NumberOfSuppliments	0.065960	0.114915	0.094889
AddressChange_Claim	0.011213	0.013065	0.000000
NumberOfCars	0.012969	0.000000	0.000000
BasePolicy	0.266603	0.095617	0.112224

	PoliceReportFiled	WitnessPresent	AgentType \
Month	0.048222	0.000000	0.018577
WeekOfMonth	0.002525	0.000000	0.000000
DayOfWeek	0.012841	0.000000	0.000000
Make	0.009755	0.000000	0.033383
AccidentArea	0.000000	0.025711	0.000000
DayOfWeekClaimed	0.010614	0.009409	0.022069
MonthClaimed	0.058189	0.000000	0.026430
WeekOfMonthClaimed	0.023282	0.000000	0.006986
Sex	0.000000	0.000000	0.008846
MaritalStatus	0.000000	0.012698	0.012194
Fault	0.025564	0.059523	0.000000
PolicyType	0.041240	0.045639	0.097221
VehicleCategory	0.038857	0.025268	0.042754
VehiclePrice	0.000000	0.000000	0.096403
Deductible	0.000000	0.000000	0.000000
Days_Policy_Accident	0.017715	0.054121	0.000000
Days_Policy_Claim	0.007099	0.000000	0.000000
PastNumberOfClaims	0.002223	0.016320	0.023476

AgeOfVehicle	0.004303	0.027724	0.013653
AgeOfPolicyHolder	0.000000	0.013846	0.000000
PoliceReportFiled	1.000000	0.195301	0.020126
WitnessPresent	0.195301	1.000000	0.000000
AgentType	0.020126	0.000000	1.000000
NumberOfSuppliments	0.018762	0.006749	0.030873
AddressChange_Claim	0.017343	0.000000	0.021645
NumberOfCars	0.022276	0.000000	0.026340
BasePolicy	0.042135	0.037662	0.082518

	NumberOfSuppliments	AddressChange_Claim	NumberOfCars \
Month	0.015619	0.007780	0.059911
WeekOfMonth	0.000000	0.000000	0.000000
DayOfWeek	0.005012	0.000000	0.000000
Make	0.050781	0.000000	0.027050
AccidentArea	0.014681	0.025268	0.006159
DayOfWeekClaimed	0.000000	0.000000	0.003409
MonthClaimed	0.015780	0.000000	0.041025
WeekOfMonthClaimed	0.009305	0.006687	0.000000
Sex	0.008177	0.000000	0.000000
MaritalStatus	0.020860	0.000000	0.012541
Fault	0.025418	0.000000	0.000000
PolicyType	0.049261	0.035023	0.019327
VehicleCategory	0.027005	0.000000	0.000000
VehiclePrice	0.054728	0.000000	0.000000
Deductible	0.011993	0.534026	0.032926
Days_Policy_Accident	0.054258	0.009411	0.049161
Days_Policy_Claim	0.036354	0.000000	0.000000
PastNumberOfClaims	0.065960	0.011213	0.012969
AgeOfVehicle	0.114915	0.013065	0.000000
AgeOfPolicyHolder	0.094889	0.000000	0.000000
PoliceReportFiled	0.018762	0.017343	0.022276
WitnessPresent	0.006749	0.000000	0.000000
AgentType	0.030873	0.021645	0.026340
NumberOfSuppliments	1.000000	0.000000	0.000000
AddressChange_Claim	0.000000	1.000000	0.467649
NumberOfCars	0.000000	0.467649	1.000000
BasePolicy	0.034918	0.004204	0.000000

	BasePolicy
Month	0.027061
WeekOfMonth	0.006613
DayOfWeek	0.043162
Make	0.114251
AccidentArea	0.055858
DayOfWeekClaimed	0.015649
MonthClaimed	0.032798

WeekOfMonthClaimed	0.013382
Sex	0.068938
MaritalStatus	0.036605
Fault	0.206502
PolicyType	0.999805
VehicleCategory	0.680729
VehiclePrice	0.204117
Deductible	0.009129
Days_Policy_Accident	0.024216
Days_Policy_Claim	0.016300
PastNumberOfClaims	0.266603
AgeOfVehicle	0.095617
AgeOfPolicyHolder	0.112224
PoliceReportFiled	0.042135
WitnessPresent	0.037662
AgentType	0.082518
NumberOfSupplements	0.034918
AddressChange_Claim	0.004204
NumberOfCars	0.000000
BasePolicy	1.000000

[27 rows x 27 columns]

Observaciones **Correlación con la variable objetivo** - Las variables categóricas muestran correlaciones más altas con el fraude.

Redundancia entre categóricas relevantes - **PolicyType** **BasePolicy** $\rightarrow V$ 0.9998 \rightarrow redundancia casi perfecta. - **PolicyType** **VehicleCategory** $\rightarrow V$ 0.9998 \rightarrow redundancia casi perfecta. - Eliminar **PolicyType** para evitar colinealidad extrema.

3.7 Descomposición de la variable objetivo por subgrupos predictivos

```
[14]: # Variables a excluir porque se van a eliminar
excluded_cols = ['FraudFound_P', 'PolicyNumber', 'RepNumber', 'PolicyType',
                'Age', 'DriverRating', 'Year']
# Variable objetivo
target_col = 'FraudFound_P'

features = [col for col in df.columns if col not in excluded_cols and not col.
            endswith('_binned')]

# Función personalizada que calcula, para cada categoría o intervalo, la tasa
# de fraude como el porcentaje de registros etiquetados como fraudulentos
def plot_fraud_rate(df, column, target='FraudFound_P', horizontal=False):
    data = df.groupby(column)[target].agg(['count', 'sum']).reset_index()
    data['percentage'] = 100 * data['sum'] / data['count']
    data = data.sort_values('percentage', ascending=False)
```

```

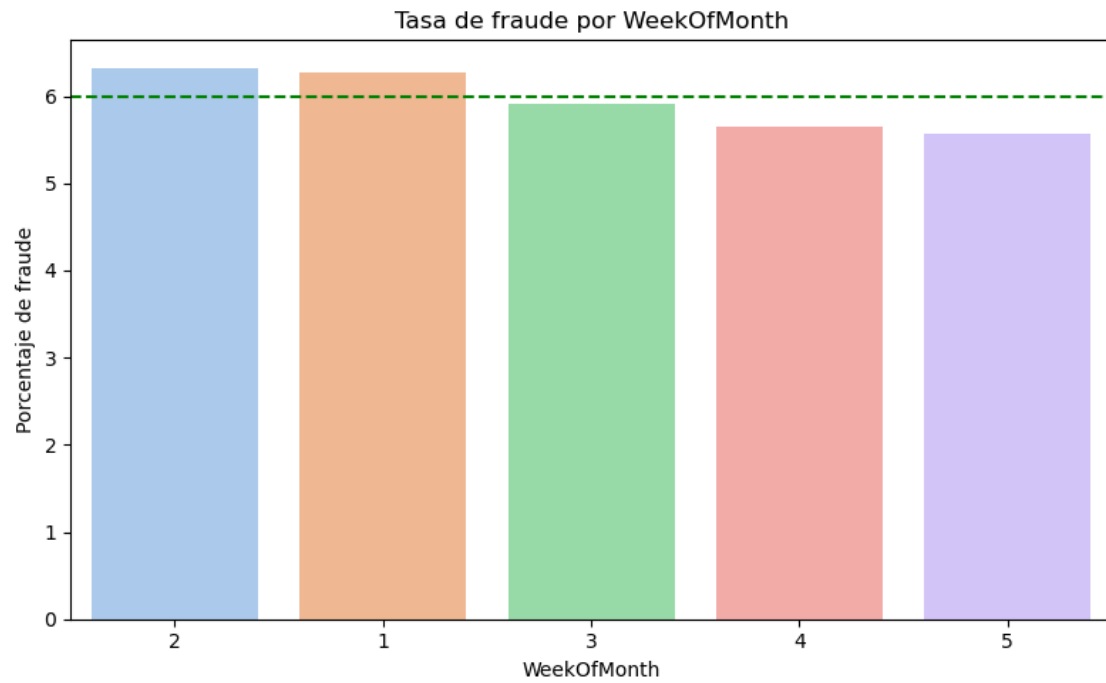
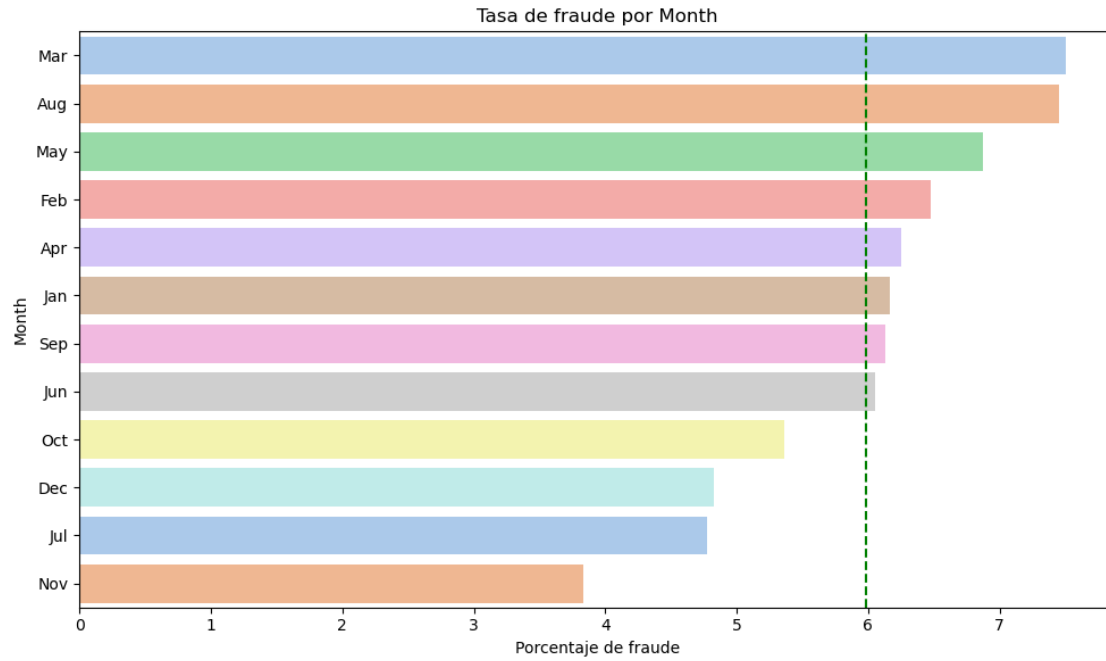
data[column] = data[column].astype(str)

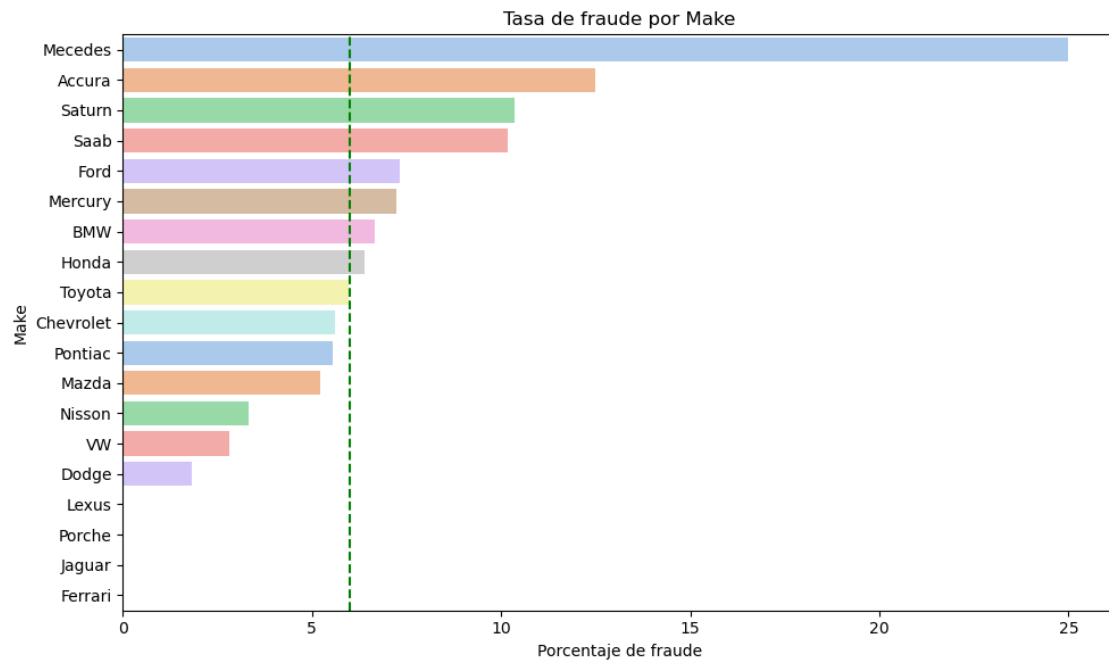
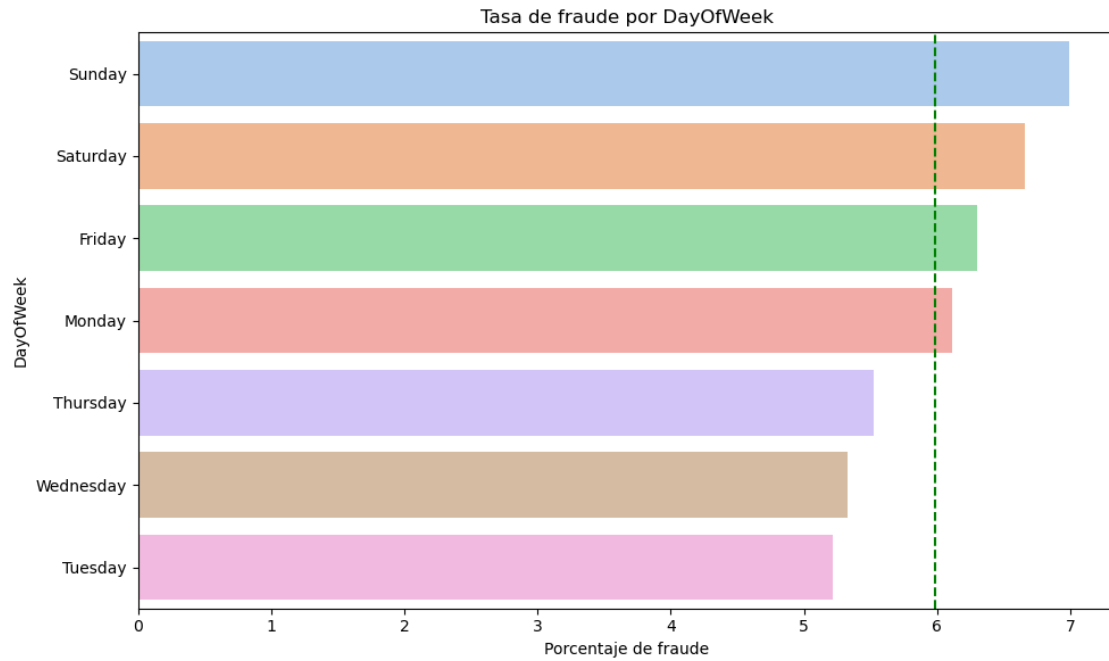
unique_vals = data[column].nunique()
palette = sns.color_palette("pastel", n_colors=unique_vals)
figsize = (10, 6) if horizontal else (8, 5)
plt.figure(figsize=figsize)
if horizontal:
    sns.barplot(y=column, x='percentage', data=data, palette=palette)
    plt.axvline(df[target].mean() * 100, color='green', linestyle='--')
    plt.xlabel('Porcentaje de fraude')
    plt.ylabel(column)
else:
    sns.barplot(x=column, y='percentage', data=data, palette=palette)
    plt.axhline(df[target].mean() * 100, color='green', linestyle='--')
    plt.ylabel('Porcentaje de fraude')
    plt.xlabel(column)

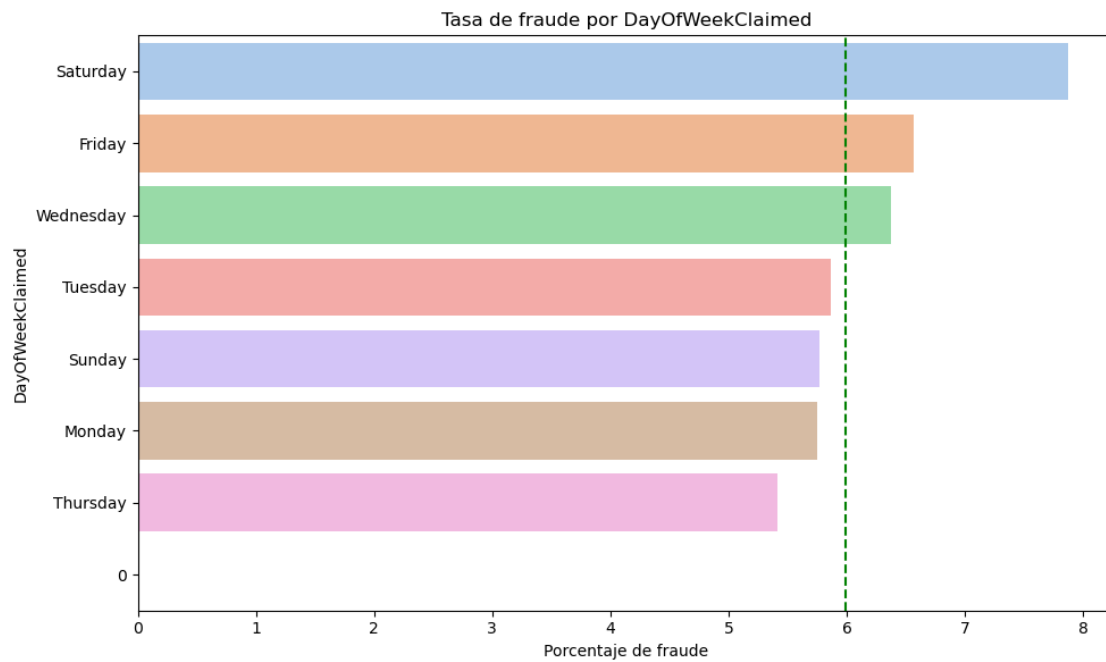
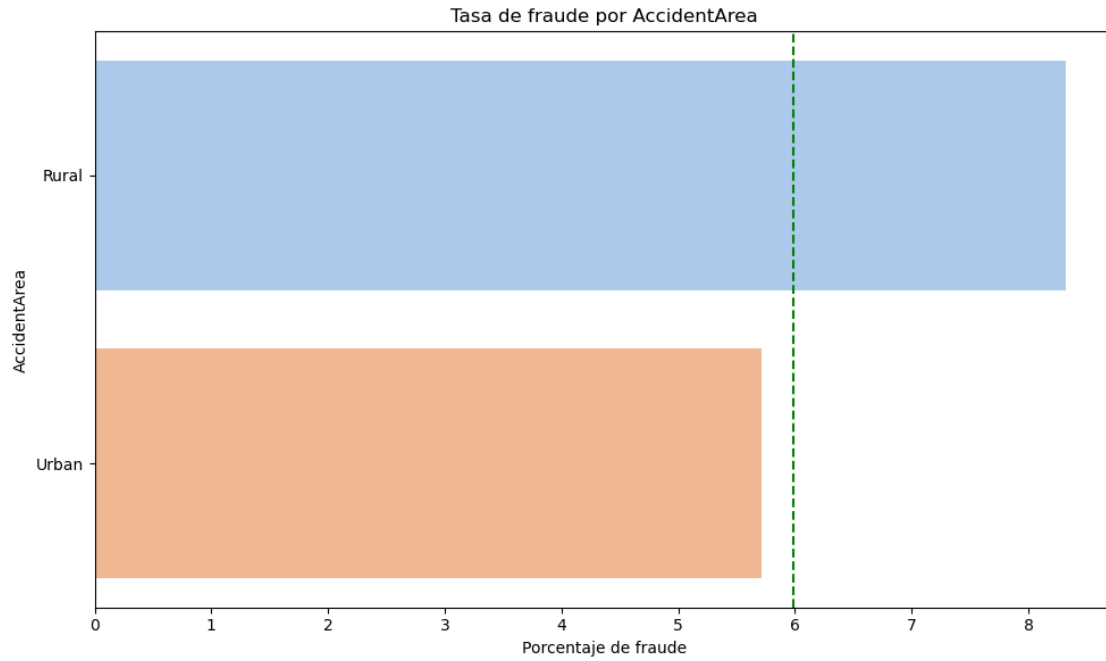
plt.title(f'Tasa de fraude por {column}', loc='center')
plt.tight_layout()
plt.show()

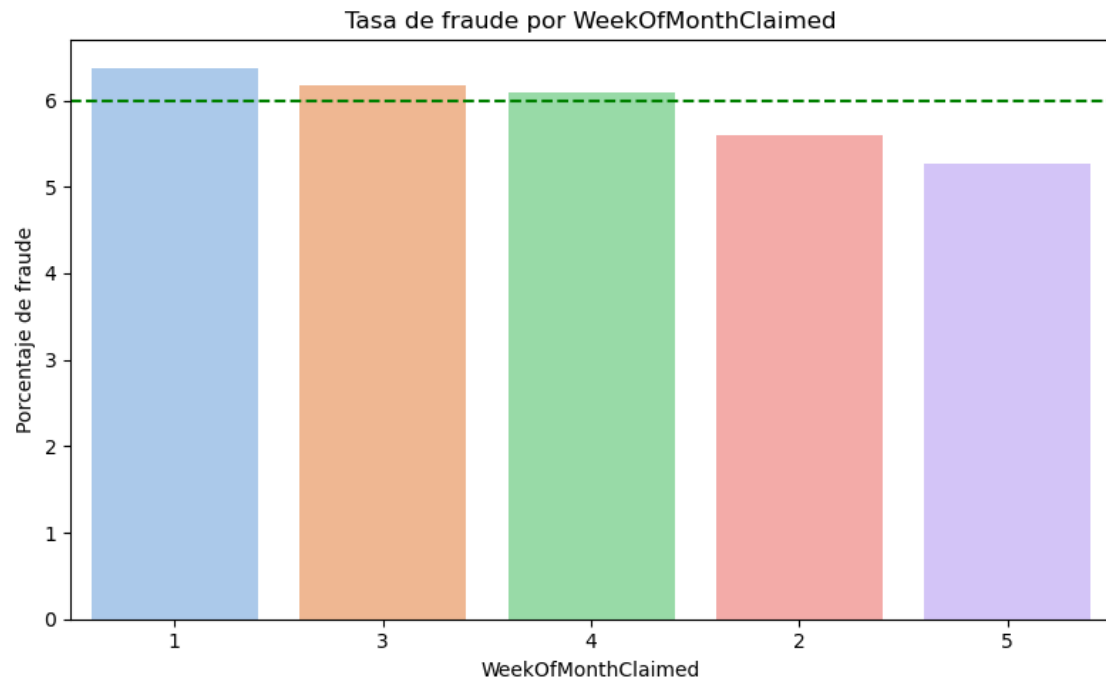
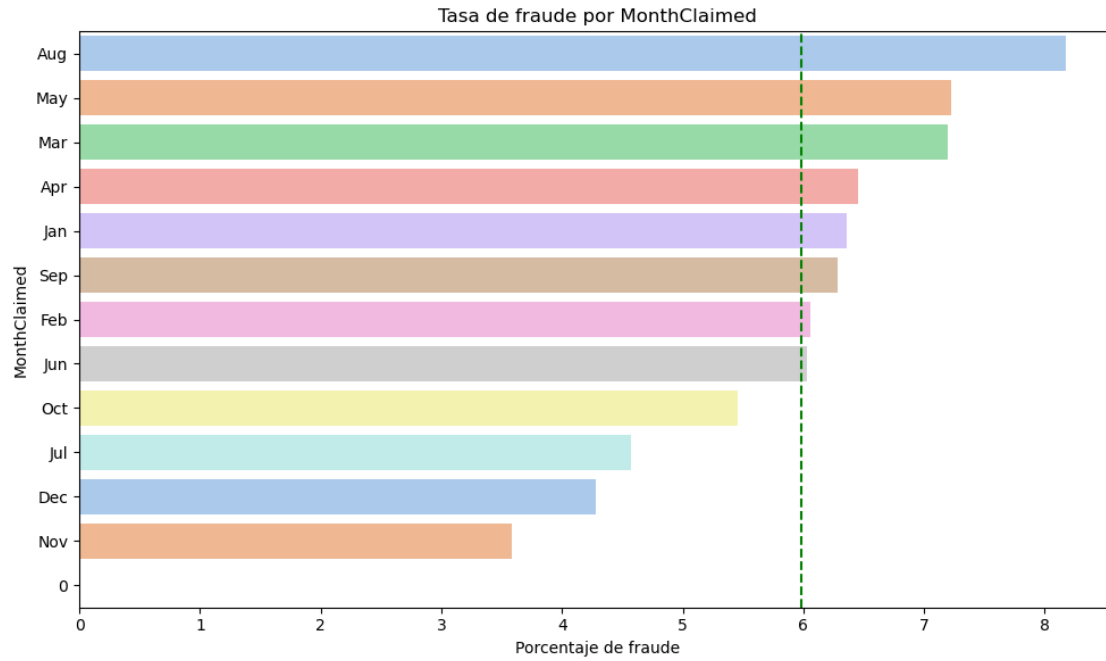
# Para variables numéricas con alta cardinalidad, proceso de discretización por
↳ cuantiles (qcut) en cinco intervalos para facilitar su interpretación y
↳ comparación.
for col in features:
    if df[col].nunique() <= 1:
        continue
    if pd.api.types.is_numeric_dtype(df[col]):
        if df[col].nunique() > 20:
            binned_col = f'{col}_binned'
            df[binned_col] = pd.qcut(df[col], q=5, duplicates='drop')
            plot_fraud_rate(df, binned_col, target=target_col, horizontal=True)
        else:
            plot_fraud_rate(df, col, target=target_col, horizontal=False)
    else:
        plot_fraud_rate(df, col, target=target_col, horizontal=True)

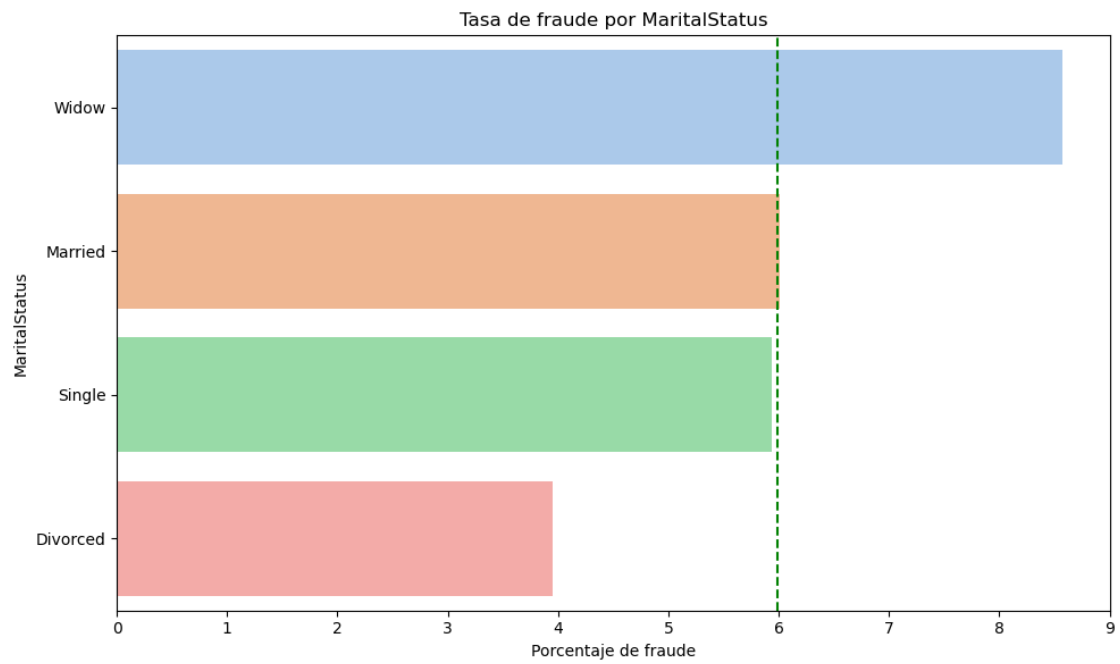
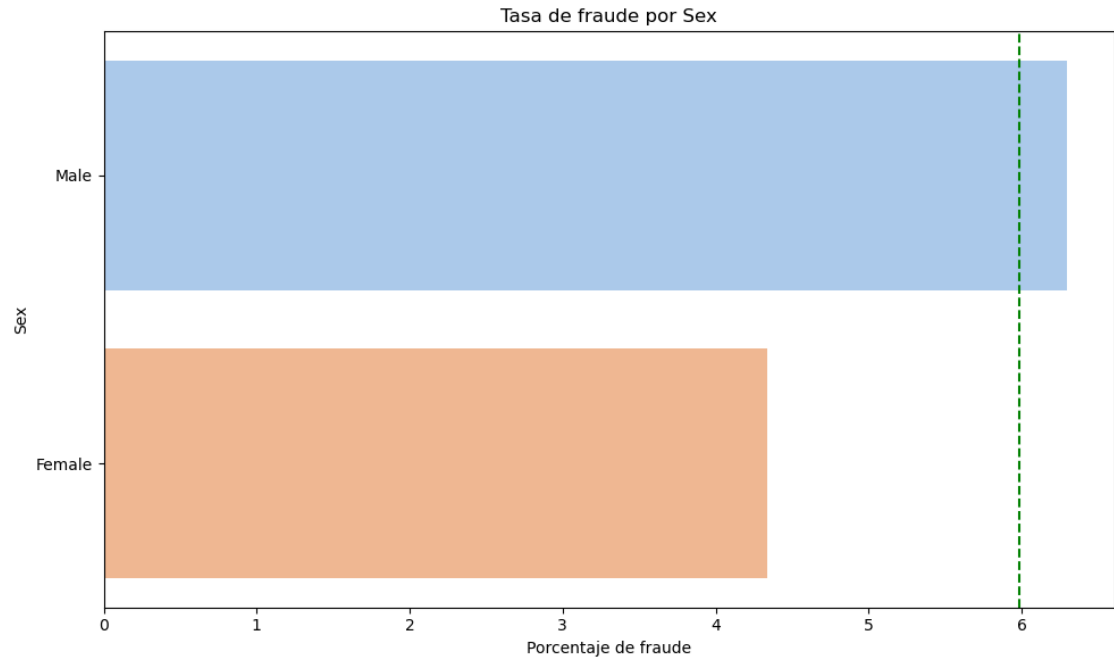
```

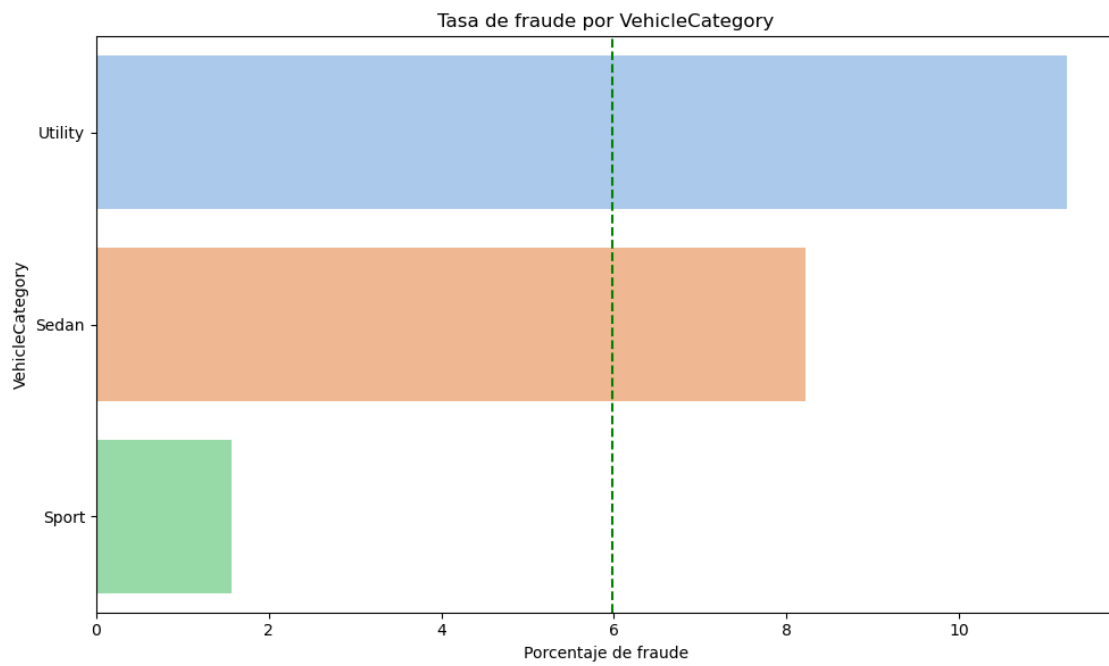
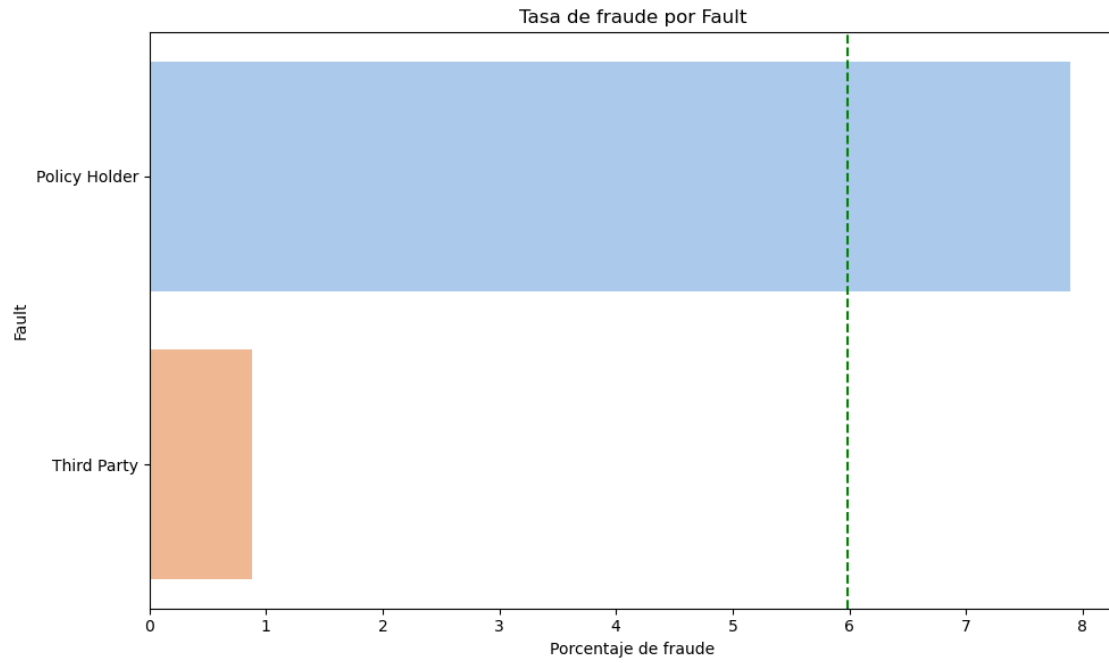


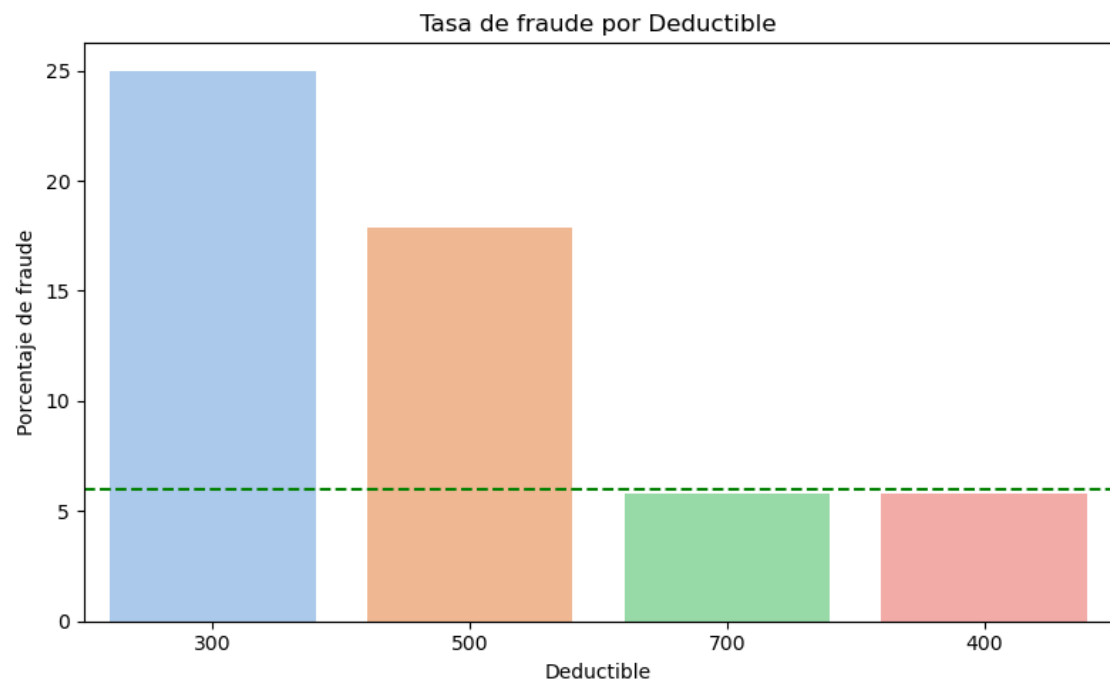
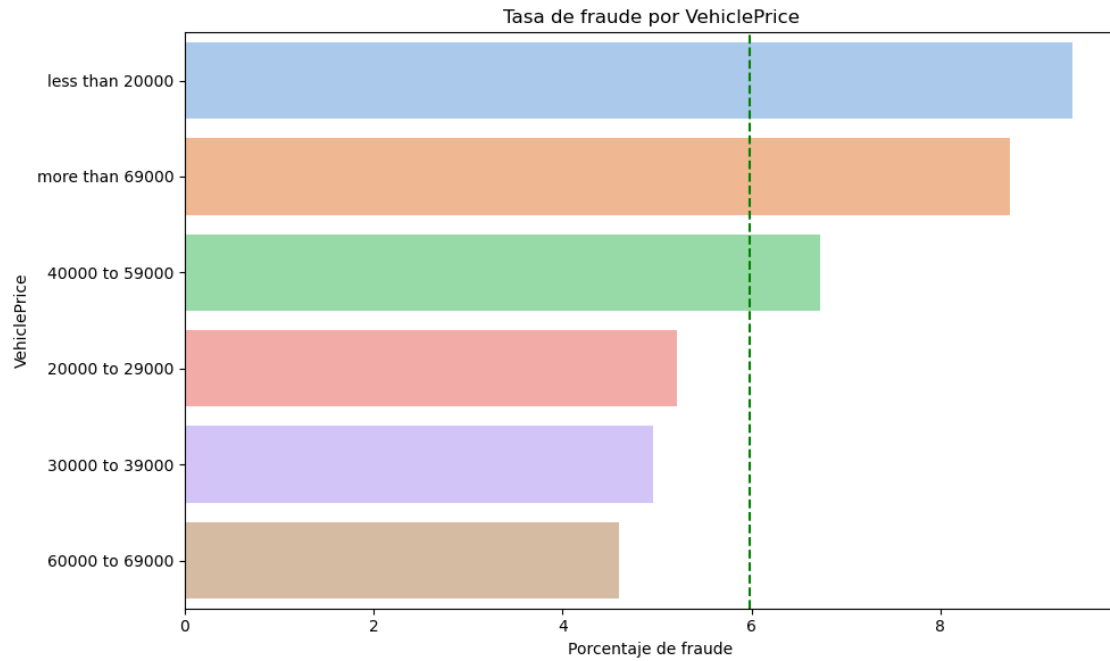


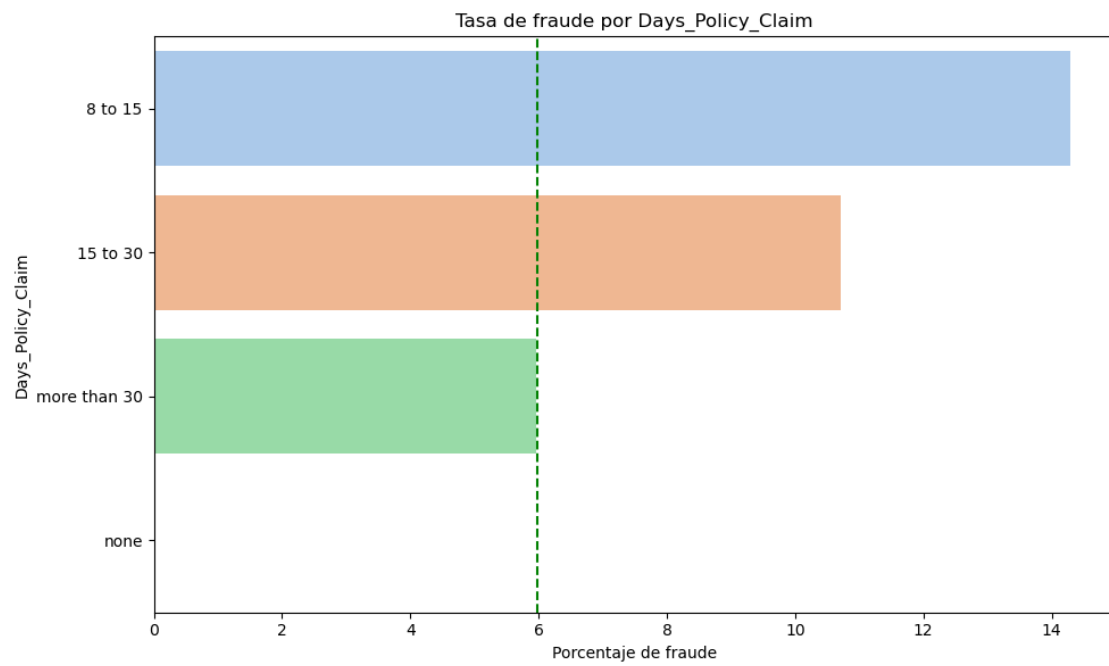
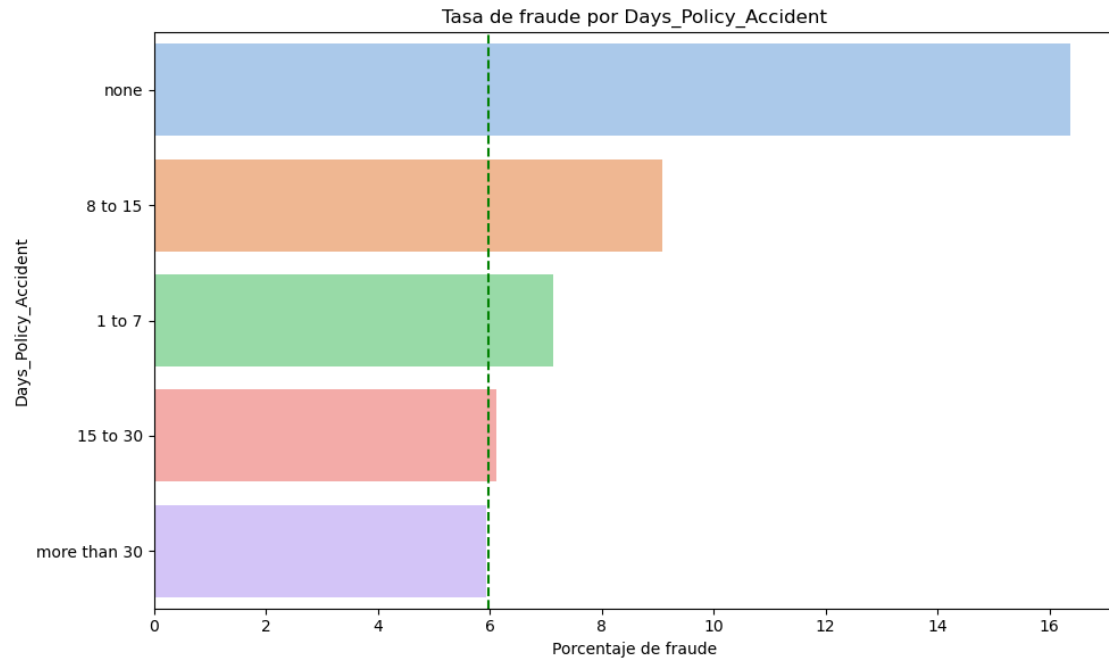


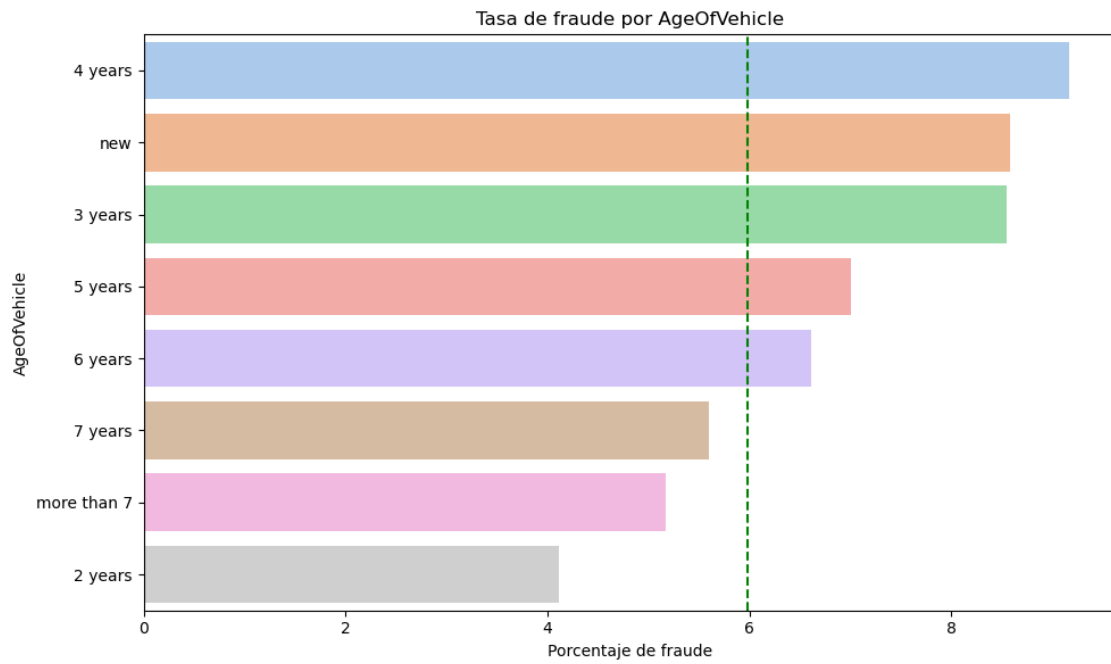
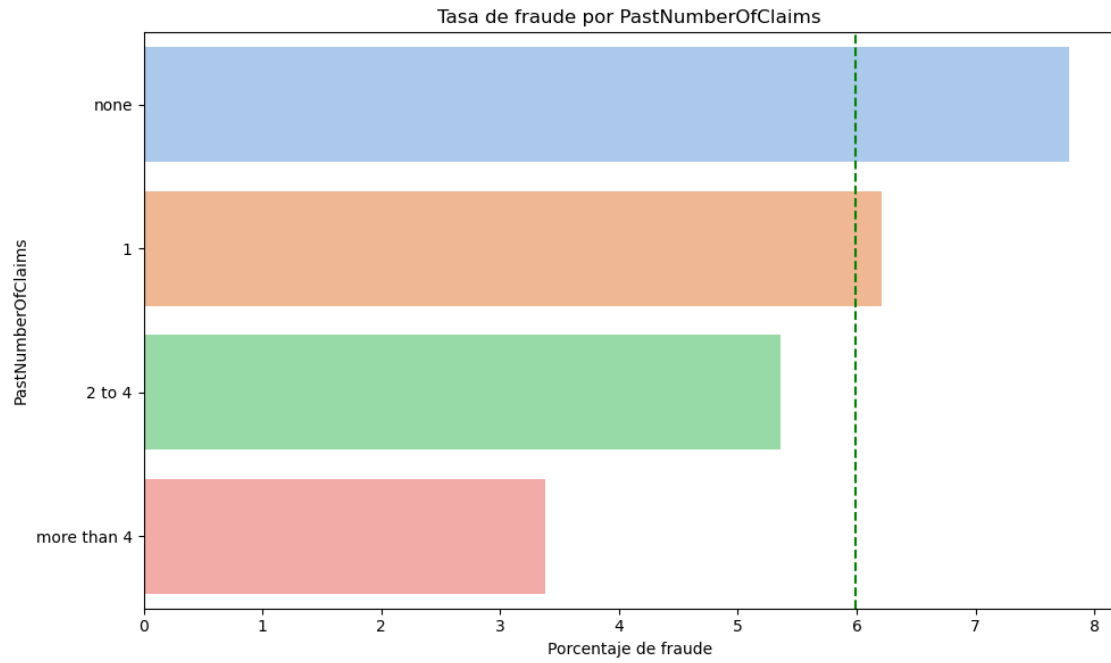


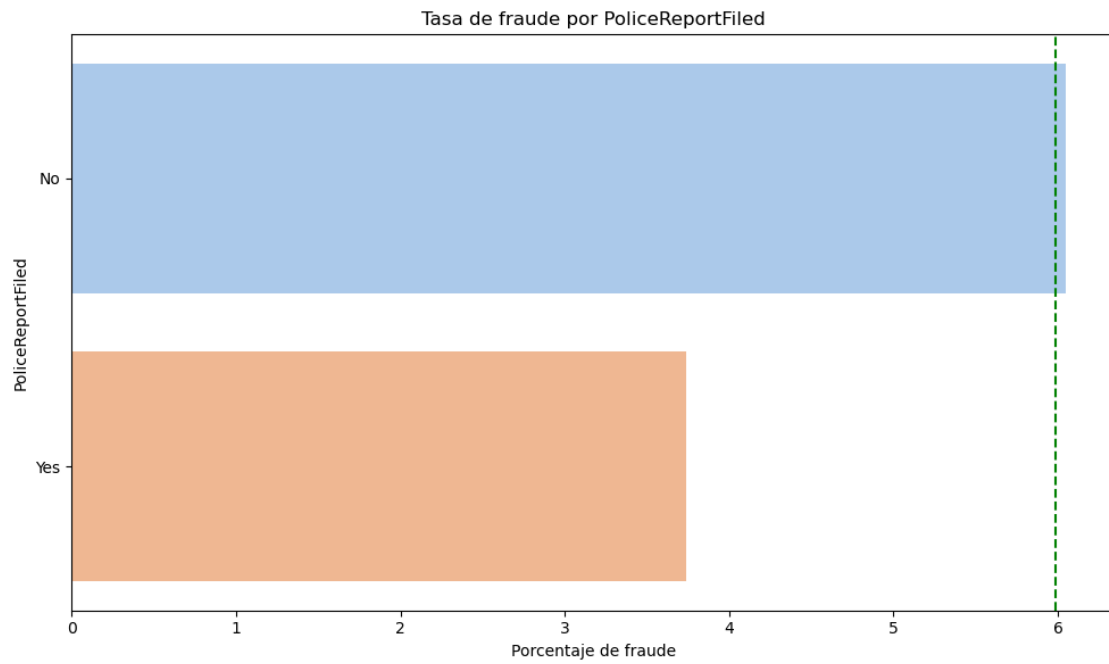
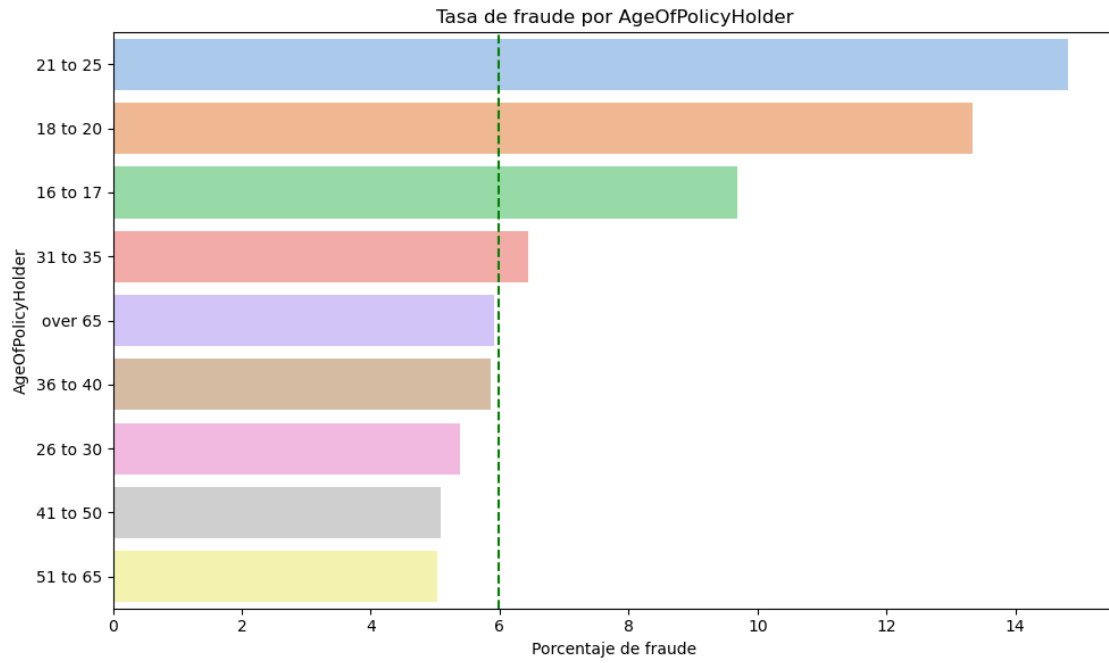


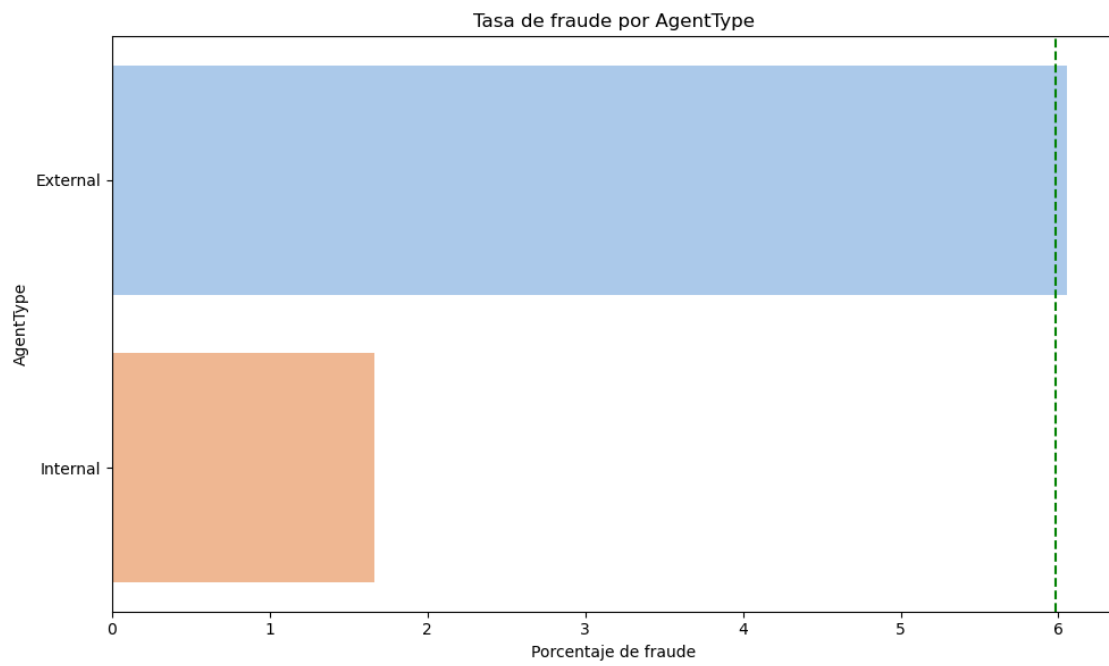
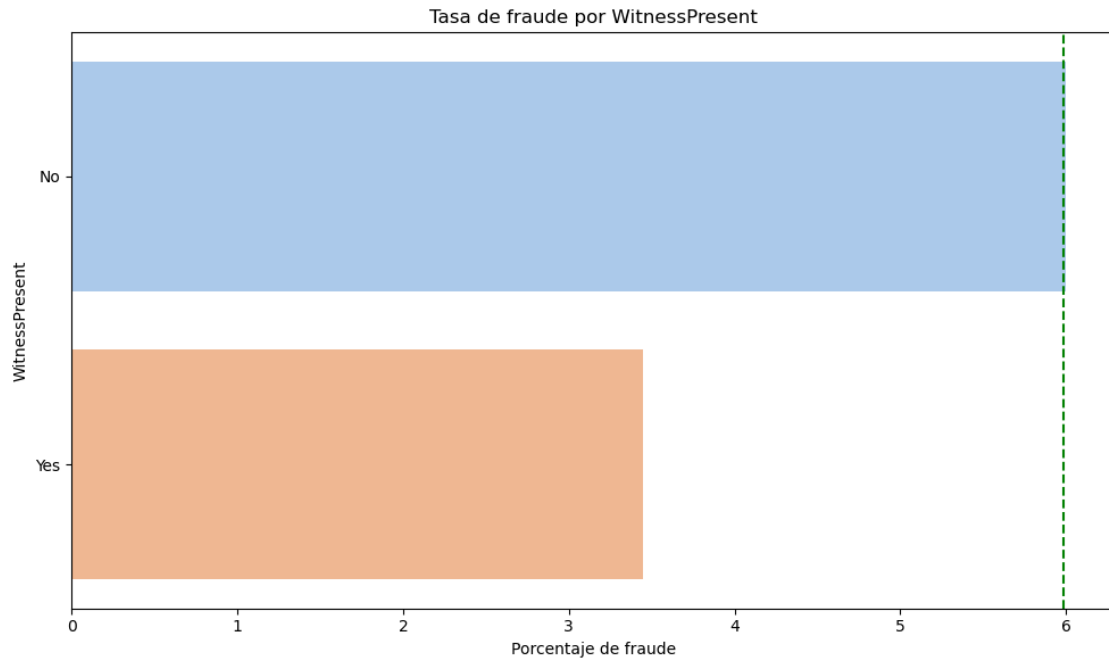


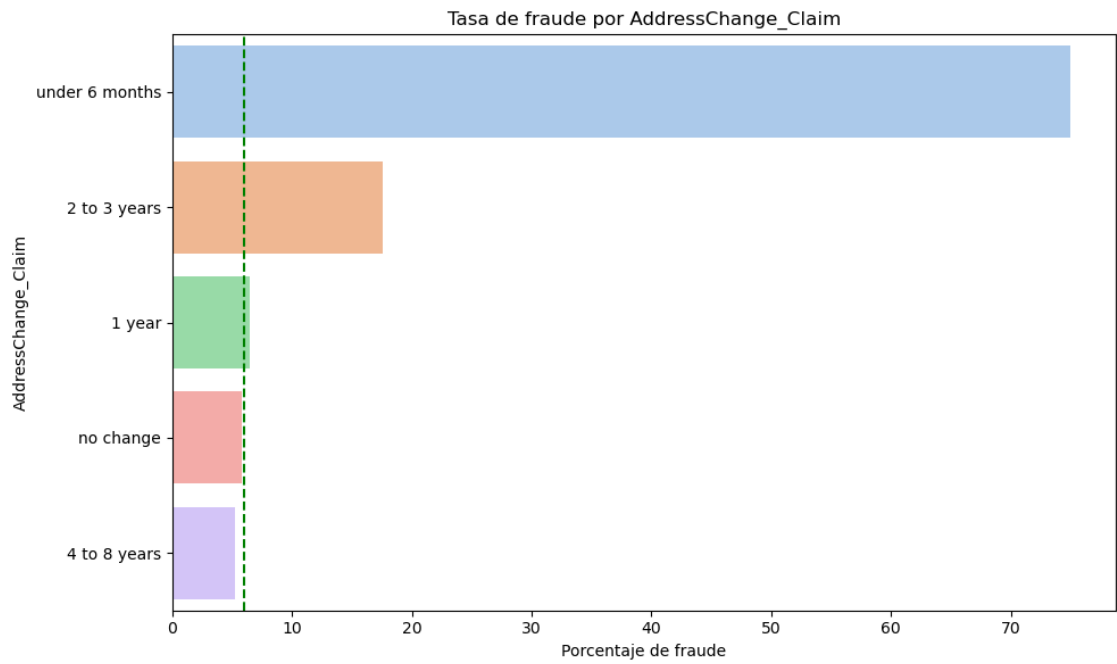
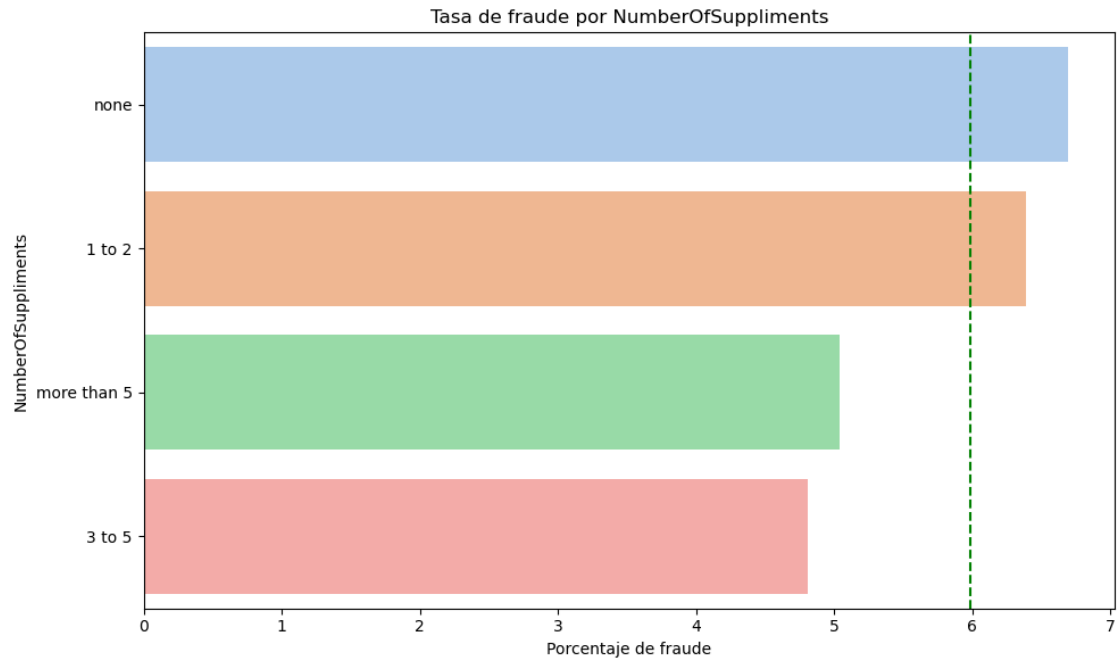


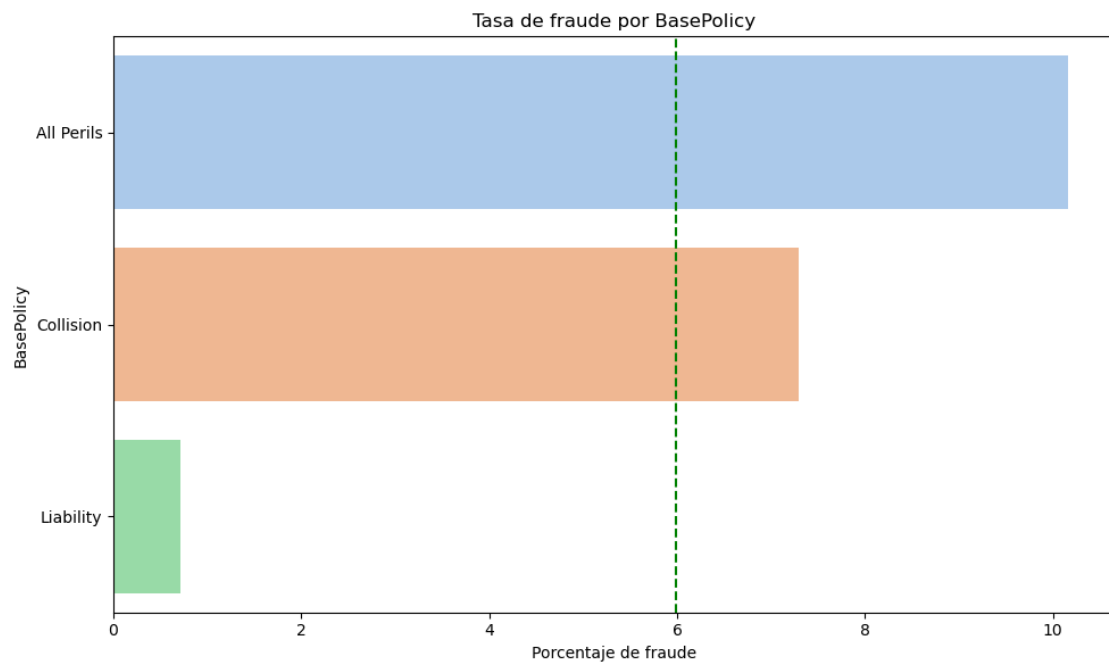
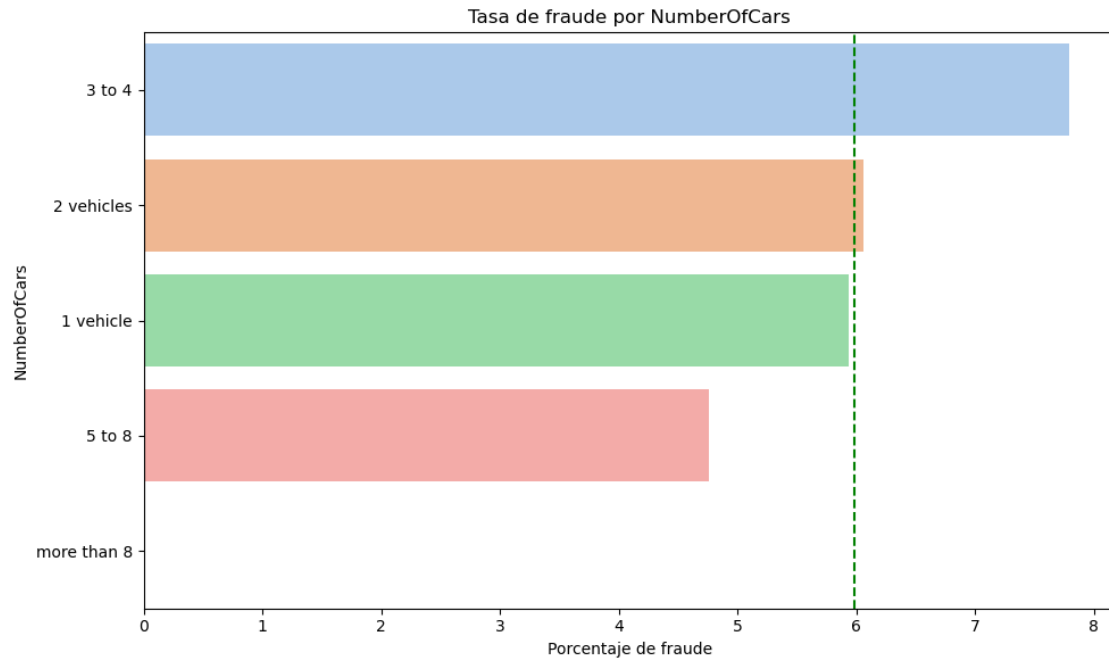












3.7.1