TC300C_Etapa 4.2-Medidas descriptivas y caracterización grafica

April 21, 2024

1 4.2 Medidas descriptivas y caracterización gráfica (Avance Evidencia 1)

Pandalytics - Equipo 1 * **A00832444** | Andrea Garza * **A01197991** | Hiram Maximiliano Muñoz Ramírez * **A00517124** | Erick Orlando Hernández Vallejo * **A01197655** | Raúl Isaí Murillo Alemán * **A01235692** | David Gerardo Martíne Hidrogo

---

Todas las variables

[48]:
```
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt

paint_per_date_df = pd.read_feather('data/paint_per_date.feather')

# Dataframe para las 20 pinturas con más litros utilizados
liters_per_paint = paint_per_date_df.groupby('paint_name')['total_liters_used'].
  ↪sum()
most_used_paints = liters_per_paint.sort_values(ascending=False).head(10).index
most_used_paints_df = paint_per_date_df.loc[most_used_paints]


# Función para identificar la frecuencia de cada valor
def do_value_counts(df: pd.DataFrame, column: str):
    print('Frecuencia de los Valores:')
    display(df['input_weight_kg'].value_counts())
    print('\n')


# Función para crear el Histomgrama
def do_histogram(df: pd.DataFrame, column: str, bin_count: int, title: str |␣
  ↪None = None, color: str | None = None):
    max = df[column].max()
    min = df[column].min()
    bin_size = (max - min) / bin_count
    print('Numero de bins: %d | Tamaño de cada bin: %.2f' % (bin_count, bin_size))
    bins = np.arange(min, max, bin_size)

    g = sns.histplot(df, x=column, bins=bins, kde=True)

    if color:
        g.get_lines()[0].set_color(color)  #se puede ecoger el color deseado
    else:
        g.get_lines()[0].set_color('black')  #color default

    mean = df[column].mean()
    std = df[column].std()
    plt.axvline(x=mean, color='red', linestyle='--')
    plt.axvline(x=mean - std, color='orange', linestyle='--')
    plt.axvline(x=mean + std, color='orange', linestyle='--')
    plt.legend(labels=['Distribución normal', 'Promedio', 'Desviación estandar'])

    plt.xticks(bins)
    plt.xticks(rotation=90)

    if title:
        plt.title(title)

    plt.show()


# Función para crear el Boxplot
def do_boxplot(data: pd.DataFrame, x, y=None, title=None, rotate=False, orient='v'):
    if y is None:
        sns.boxplot(x=data[x])
    else:
        sns.boxplot(data, x=x, y=y, hue=x, palette='Spectral', orient=orient)

    if rotate:
        plt.xticks(rotation=90)

    if title:
        plt.title(title)
    plt.show()

# Ejemplo de uso:
# do_boxplot(paint_per_date_df, x='production_line', y='total_liters_used', title='production_line vs. total_liters_used')
# sns.boxplot(coating_df, x=column, y=y).set(title=' %s vs. %s' % (column, y))
# plt.show()

%matplotlib inline
sns.color_palette("Spectral", as_cmap=True)
```

[48]:

[49]:

```
paint_per_date_df
```

[49]:

```
length_m  \
paint_name                    date       production_line user
0001-PRIMER 4457              2022-01-16 Pintado 2       ALEINSUMOS
2193.000000
                              2022-01-17 Pintado 2       ALEINSUMOS
5415.000000
                              2022-01-20 Pintado 1       NaN
9803.000000
                              2022-01-21 Pintado 1       ALEINSUMOS
3553.000000
                              2022-01-22 Pintado 1       ALEINSUMOS
4024.000000
…
…
2453-GRAY BACKER EDGE         2023-08-21 Pintado 1       NaN
1308.000000
                                         Pintado 2       ALEINSUMOS
51680.000000
2470-HG GRAY POLYESTER BACKER 2022-06-12 Pintado 2       ALEINSUMOS
33450.924101
                              2022-07-24 Pintado 2       ALEINSUMOS
10482.000000
                              2022-08-30 Pintado 2       ALEINSUMOS
1212.000000

m2  \
paint_name                    date       production_line user
0001-PRIMER 4457              2022-01-16 Pintado 2       ALEINSUMOS
2616.249000
                              2022-01-17 Pintado 2       ALEINSUMOS
4972.966000
                              2022-01-20 Pintado 1       NaN
```

```
12378.306000
                              2022-01-21 Pintado 1       ALEINSUMOS
3985.378000
                              2022-01-22 Pintado 1       ALEINSUMOS
3399.081000
…
…
2453-GRAY BACKER EDGE         2023-08-21 Pintado 1       NaN
1278.516000
                                         Pintado 2       ALEINSUMOS
43799.101917
2470-HG GRAY POLYESTER BACKER 2022-06-12 Pintado 2       ALEINSUMOS
32844.694259
                              2022-07-24 Pintado 2       ALEINSUMOS
10166.061000
                              2022-08-30 Pintado 2       ALEINSUMOS
1481.064000

input_weight_kg  \
paint_name                    date       production_line user
0001-PRIMER 4457              2022-01-16 Pintado 2       ALEINSUMOS
48915.00000
                              2022-01-17 Pintado 2       ALEINSUMOS
78318.00000
                              2022-01-20 Pintado 1       NaN
120892.00000
                              2022-01-21 Pintado 1       ALEINSUMOS
43759.00000
                              2022-01-22 Pintado 1       ALEINSUMOS
40393.00000
…
…
2453-GRAY BACKER EDGE         2023-08-21 Pintado 1       NaN
14305.00000
                                         Pintado 2       ALEINSUMOS
863001.00000
2470-HG GRAY POLYESTER BACKER 2022-06-12 Pintado 2       ALEINSUMOS
382250.54132
                              2022-07-24 Pintado 2       ALEINSUMOS
150833.00000
                              2022-08-30 Pintado 2       ALEINSUMOS
21800.00000

weight_kg  \
paint_name                    date       production_line user
0001-PRIMER 4457              2022-01-16 Pintado 2       ALEINSUMOS
16060.000000
```

```
                              2022-01-17 Pintado 2       ALEINSUMOS
23810.000000
                              2022-01-20 Pintado 1       NaN
72622.000000
                              2022-01-21 Pintado 1       ALEINSUMOS
23425.000000
                              2022-01-22 Pintado 1       ALEINSUMOS
20327.000000
…
…
2453-GRAY BACKER EDGE         2023-08-21 Pintado 1       NaN
6109.000000
                                         Pintado 2       ALEINSUMOS
420438.000000
2470-HG GRAY POLYESTER BACKER 2022-06-12 Pintado 2       ALEINSUMOS
161899.807635
                              2022-07-24 Pintado 2       ALEINSUMOS
57846.000000
                              2022-08-30 Pintado 2       ALEINSUMOS
7170.000000

avg_thickness_mm  \
paint_name                    date       production_line user
0001-PRIMER 4457              2022-01-16 Pintado 2       ALEINSUMOS
0.770000
                              2022-01-17 Pintado 2       ALEINSUMOS
0.602967
                              2022-01-20 Pintado 1       NaN
0.732908
                              2022-01-21 Pintado 1       ALEINSUMOS
0.727200
                              2022-01-22 Pintado 1       ALEINSUMOS
0.737667
…
…
2453-GRAY BACKER EDGE         2023-08-21 Pintado 1       NaN
0.600667
                                         Pintado 2       ALEINSUMOS
1.209975
2470-HG GRAY POLYESTER BACKER 2022-06-12 Pintado 2       ALEINSUMOS
0.659627
                              2022-07-24 Pintado 2       ALEINSUMOS
0.709167
                              2022-08-30 Pintado 2       ALEINSUMOS
0.605000

total_liters_used  \
paint_name                    date       production_line user
0001-PRIMER 4457              2022-01-16 Pintado 2       ALEINSUMOS
70.00
                              2022-01-17 Pintado 2       ALEINSUMOS
70.00
                              2022-01-20 Pintado 1       NaN
NaN
                              2022-01-21 Pintado 1       ALEINSUMOS
400.00
                              2022-01-22 Pintado 1       ALEINSUMOS
300.00
…
…
2453-GRAY BACKER EDGE         2023-08-21 Pintado 1       NaN
NaN
                                         Pintado 2       ALEINSUMOS
800.00
2470-HG GRAY POLYESTER BACKER 2022-06-12 Pintado 2       ALEINSUMOS
218.25
                              2022-07-24 Pintado 2       ALEINSUMOS
160.25
                              2022-08-30 Pintado 2       ALEINSUMOS
59.00

monetary_value_usd  \
paint_name                    date       production_line user
0001-PRIMER 4457              2022-01-16 Pintado 2       ALEINSUMOS
415.80
                              2022-01-17 Pintado 2       ALEINSUMOS
415.80
                              2022-01-20 Pintado 1       NaN
NaN
                              2022-01-21 Pintado 1       ALEINSUMOS
2376.00
                              2022-01-22 Pintado 1       ALEINSUMOS
1782.00
…
…
2453-GRAY BACKER EDGE         2023-08-21 Pintado 1       NaN
NaN
                                         Pintado 2       ALEINSUMOS
7656.00
2470-HG GRAY POLYESTER BACKER 2022-06-12 Pintado 2       ALEINSUMOS
2490.23
                              2022-07-24 Pintado 2       ALEINSUMOS
1828.45
                              2022-08-30 Pintado 2       ALEINSUMOS
```

```
673.19

expected_yield  \
paint_name                    date       production_line user
0001-PRIMER 4457              2022-01-16 Pintado 2       ALEINSUMOS
58.267717
                              2022-01-17 Pintado 2       ALEINSUMOS
58.267717
                              2022-01-20 Pintado 1       NaN
58.267717
                              2022-01-21 Pintado 1       ALEINSUMOS
58.267717
                              2022-01-22 Pintado 1       ALEINSUMOS
58.267717
…
…
2453-GRAY BACKER EDGE         2023-08-21 Pintado 1       NaN
70.866142
                                         Pintado 2       ALEINSUMOS
70.866142
2470-HG GRAY POLYESTER BACKER 2022-06-12 Pintado 2       ALEINSUMOS
81.102362
                              2022-07-24 Pintado 2       ALEINSUMOS
81.102362
                              2022-08-30 Pintado 2       ALEINSUMOS
81.102362

                                                                     real_yield
\
paint_name                    date       production_line user
0001-PRIMER 4457              2022-01-16 Pintado 2       ALEINSUMOS   37.374986
                              2022-01-17 Pintado 2       ALEINSUMOS   71.042371
                              2022-01-20 Pintado 1       NaN                NaN
                              2022-01-21 Pintado 1       ALEINSUMOS    9.963445
                              2022-01-22 Pintado 1       ALEINSUMOS   11.330270
…                                                                         …
2453-GRAY BACKER EDGE         2023-08-21 Pintado 1       NaN                NaN
                                         Pintado 2       ALEINSUMOS   54.748877
2470-HG GRAY POLYESTER BACKER 2022-06-12 Pintado 2       ALEINSUMOS  150.491154
                              2022-07-24 Pintado 2       ALEINSUMOS   63.438758
                              2022-08-30 Pintado 2       ALEINSUMOS   25.102780

yield_difference
paint_name                    date       production_line user
0001-PRIMER 4457              2022-01-16 Pintado 2       ALEINSUMOS
20.892731
                              2022-01-17 Pintado 2       ALEINSUMOS
```

```
12.774655
                              2022-01-20 Pintado 1       NaN
NaN
                              2022-01-21 Pintado 1       ALEINSUMOS
48.304272
                              2022-01-22 Pintado 1       ALEINSUMOS
46.937447
…
…
2453-GRAY BACKER EDGE         2023-08-21 Pintado 1       NaN
NaN
                                         Pintado 2       ALEINSUMOS
16.117264
2470-HG GRAY POLYESTER BACKER 2022-06-12 Pintado 2       ALEINSUMOS
69.388791
                              2022-07-24 Pintado 2       ALEINSUMOS
17.663604
                              2022-08-30 Pintado 2       ALEINSUMOS
55.999583

[3872 rows x 10 columns]
```

1.1 Variables cuantitativas

[50]:

```python
# Tipo de datos
print('Data Type:\n', paint_per_date_df.dtypes)

print('\nNo hay variables discretas\n')

#Variables Continuas
unique_values = paint_per_date_df.nunique()
discrete_variables = [column for column in paint_per_date_df.columns if
                      unique_values[column] < len(paint_per_date_df) / 0.05]
print("\nVariables Continuas:")
for variable in discrete_variables:
    print(variable)

print(paint_per_date_df.describe())
```

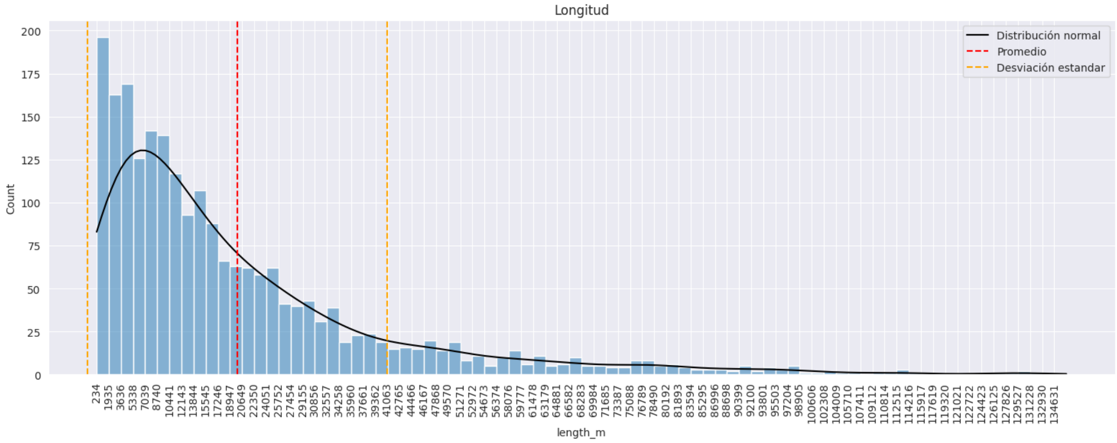
```
Data Type:
 length_m              float64
m2                    float64
input_weight_kg       float64
weight_kg             float64
avg_thickness_mm      float64
total_liters_used     float64
monetary_value_usd    float64
expected_yield        float64
```

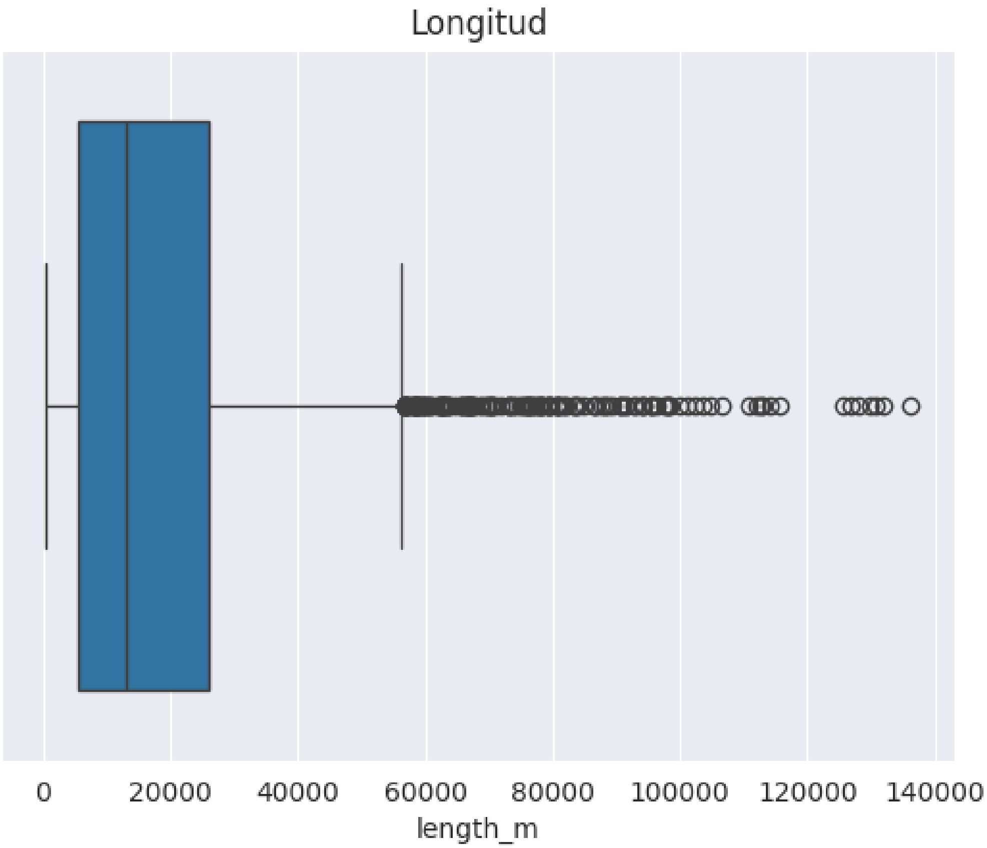
```
real_yield            float64
yield_difference      float64
dtype: object

No hay variables discretas


Variables Continuas:
length_m
m2
input_weight_kg
weight_kg
avg_thickness_mm
total_liters_used
monetary_value_usd
expected_yield
real_yield
yield_difference
            length_m             m2  input_weight_kg      weight_kg  \
count    3872.000000    3872.000000     3.872000e+03    3872.000000   
mean    17918.959565   18486.555737     2.141211e+05   84158.091498   
std     19917.802468   19427.162674     2.145114e+05   84317.029552   
min       222.000000     191.808000     2.010000e+03    1072.000000   
25%      4423.974053    4763.643250     5.571750e+04   21390.250000   
50%     10963.000000   11942.166500     1.438055e+05   55176.500000   
75%     23771.000000   24961.764250     3.044788e+05  117657.750000   
max    136332.000000  124782.771000     1.289409e+06  540188.597225   

       avg_thickness_mm  total_liters_used  monetary_value_usd  \
count       3872.000000        2973.000000         2973.000000   
mean           0.619018         431.347683         3099.855271   
std            0.187768         361.007546         2770.162029   
min            0.352700           5.000000           33.000000   
25%            0.461785         160.000000         1107.600000   
50%            0.585528         330.000000         2242.500000   
75%            0.723700         600.000000         4310.400000   
max            1.231100        2031.500000        17436.520000   

       expected_yield   real_yield  yield_difference  
count     3870.000000  2973.000000       2973.000000  
mean        53.222596    58.337904         31.950035  
std         15.802120    56.714807         50.110225  
min         18.637993     0.566530          0.006838  
25%         44.291339    32.501466          8.820625  
50%         54.073034    46.973794         20.790847  
75%         67.637795    66.873483         39.554919  
max         97.894737  1392.841000       1326.699268  
```

1.1.1 length_m

```
[51]: # Histograma
      plt.figure(figsize=(18, 6))
      do_histogram(most_used_paints_df, 'length_m', bin_count=80, title='Longitud')
      do_boxplot(most_used_paints_df, 'length_m', title='Longitud')
```

```
Numero de bins: 80 | Tamaño de cada bin: 1701.22
```

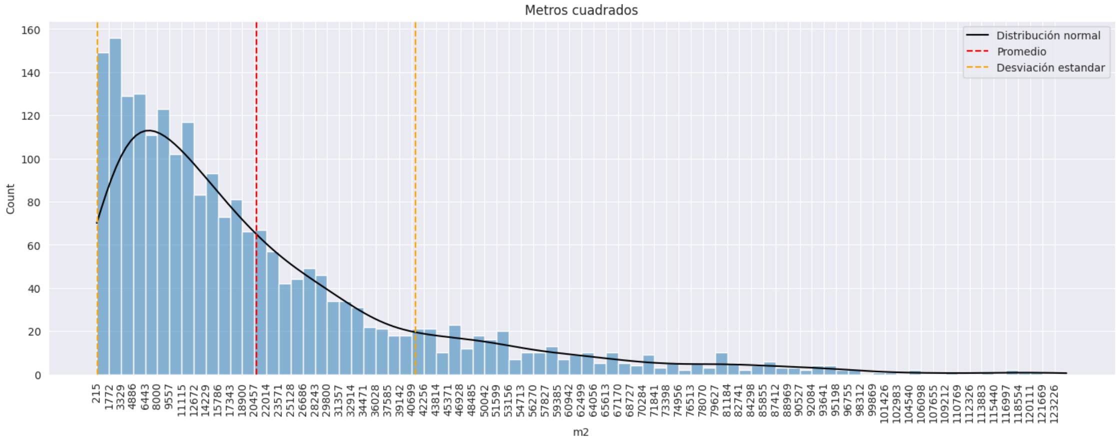


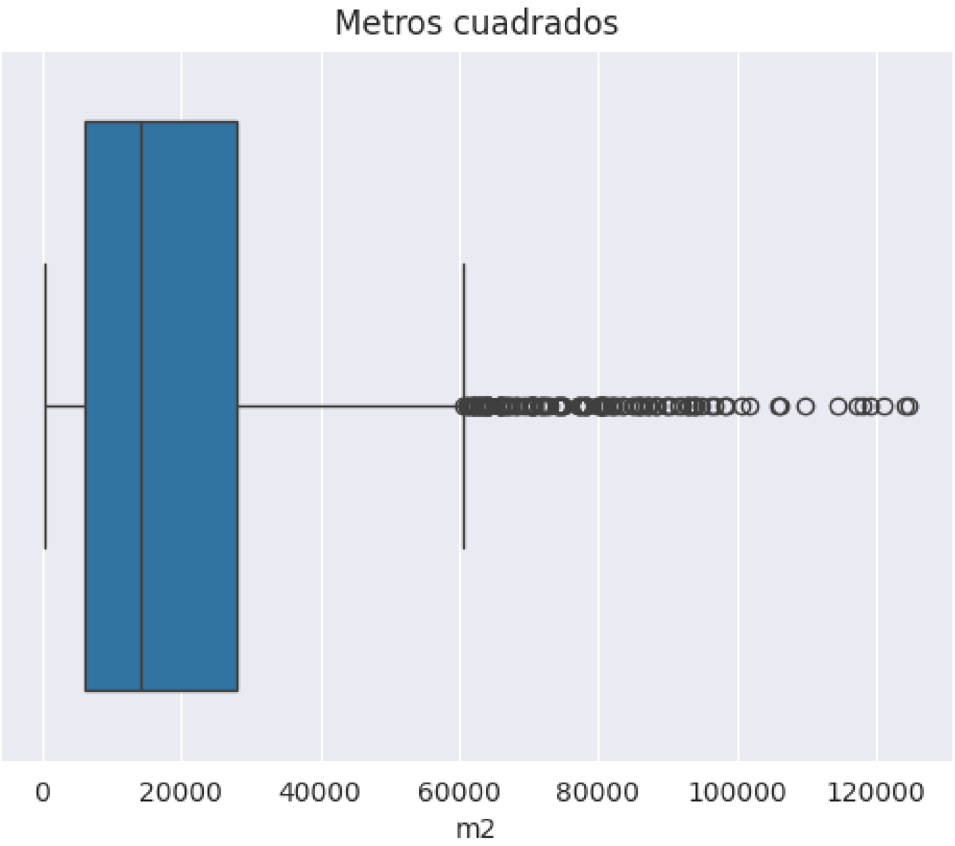


1.1.2 m2

```
[52]: # Histograma
      plt.figure(figsize=(18, 6))
      do_histogram(most_used_paints_df, 'm2', 80, title='Metros cuadrados', )
      do_boxplot(most_used_paints_df, 'm2', title='Metros cuadrados')
```

```
Numero de bins: 80 | Tamaño de cada bin: 1557.10
```

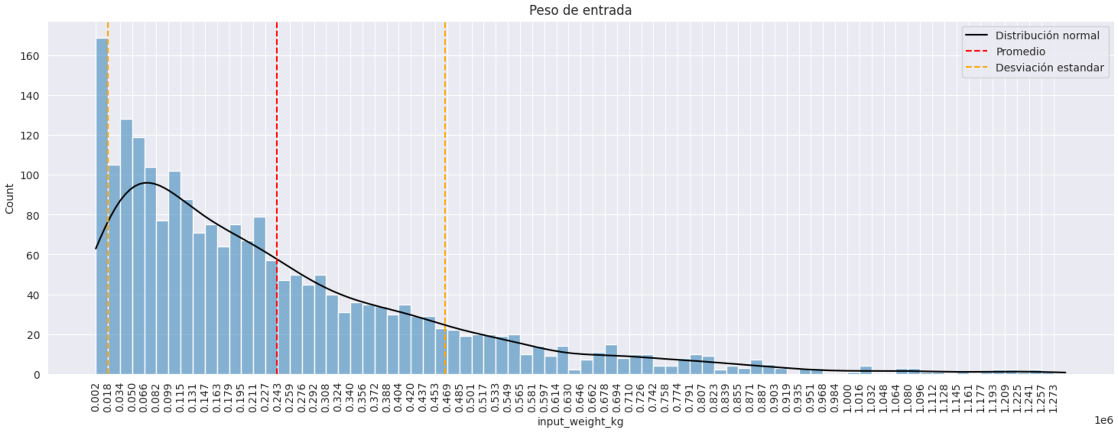


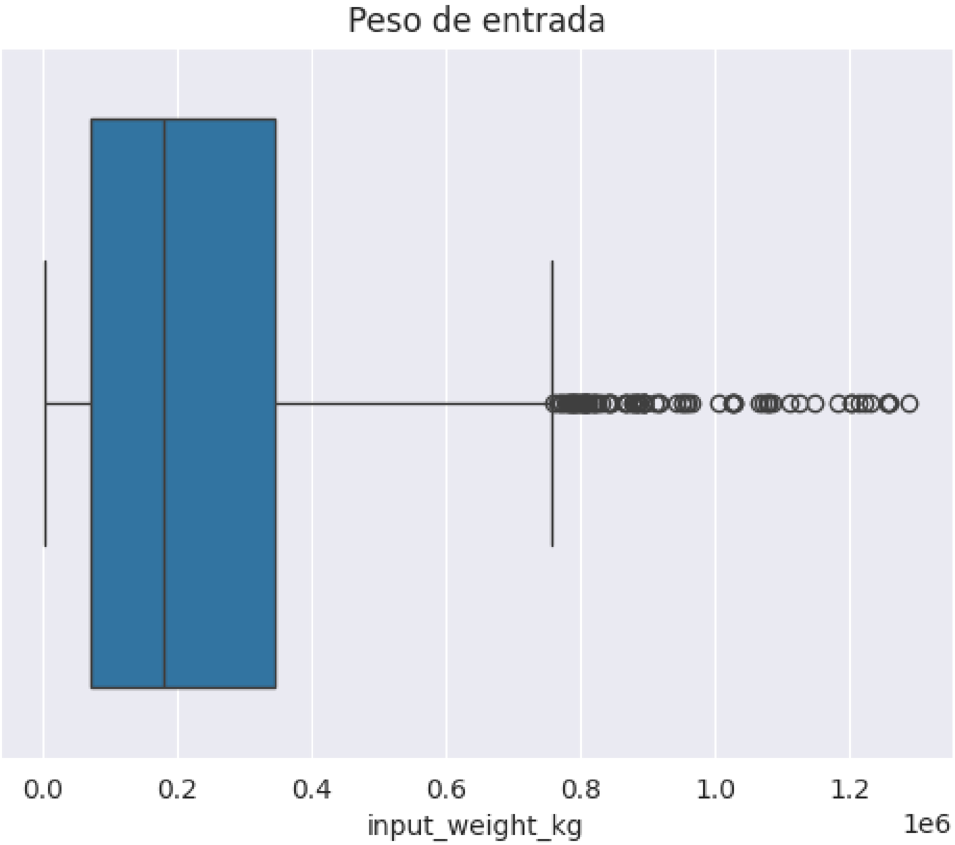


1.1.3 input_weight_kg

```
[53]: # Histograma
plt.figure(figsize=(18, 6))
do_histogram(most_used_paints_df, 'input_weight_kg', 80, title='Peso de␣
  ↪entrada')

# Boxplot
do_boxplot(most_used_paints_df, 'input_weight_kg', title='Peso de entrada')
```

```
Numero de bins: 80 | Tamaño de cada bin: 16092.49
```

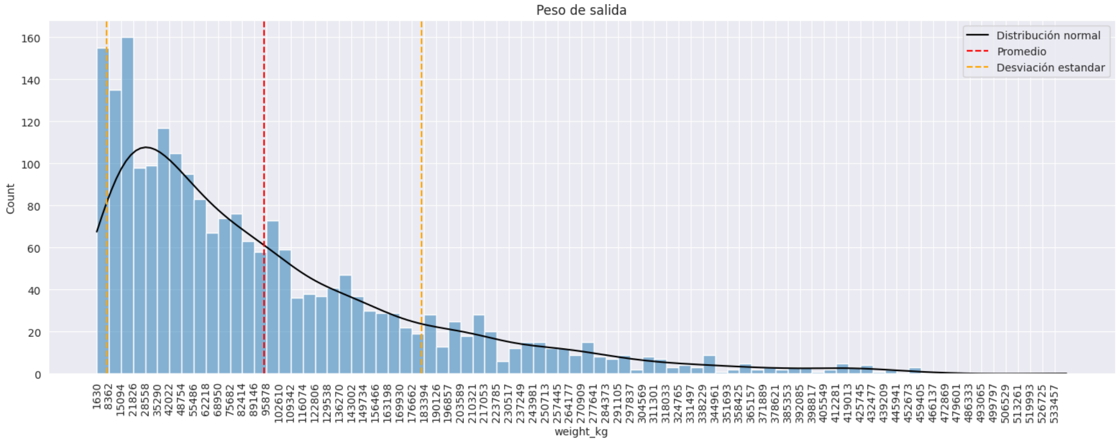

Peso de entrada
Distribución normal
Promedio
Desviación estandar
Count
160
140
120
100
80
60
40
20
0
0.002
0.018
0.034
0.050
0.066
0.082
0.099
0.115
0.131
0.147
0.163
0.179
0.195
0.211
0.227
0.243
0.259
0.276
0.292
0.308
0.324
0.340
0.356
0.372
0.388
0.404
0.420
0.437
0.453
0.469
0.485
0.501
0.517
0.533
0.549
0.565
0.581
0.597
0.614
0.630
0.646
0.662
0.678
0.694
0.710
0.726
0.742
0.758
0.774
0.791
0.807
0.823
0.839
0.855
0.871
0.887
0.903
0.919
0.935
0.951
0.968
0.984
1.000
1.016
1.032
1.048
1.064
1.080
1.096
1.112
1.128
1.145
1.161
1.177
1.193
1.209
1.225
1.241
1.257
1.273
input_weight_kg
1e6

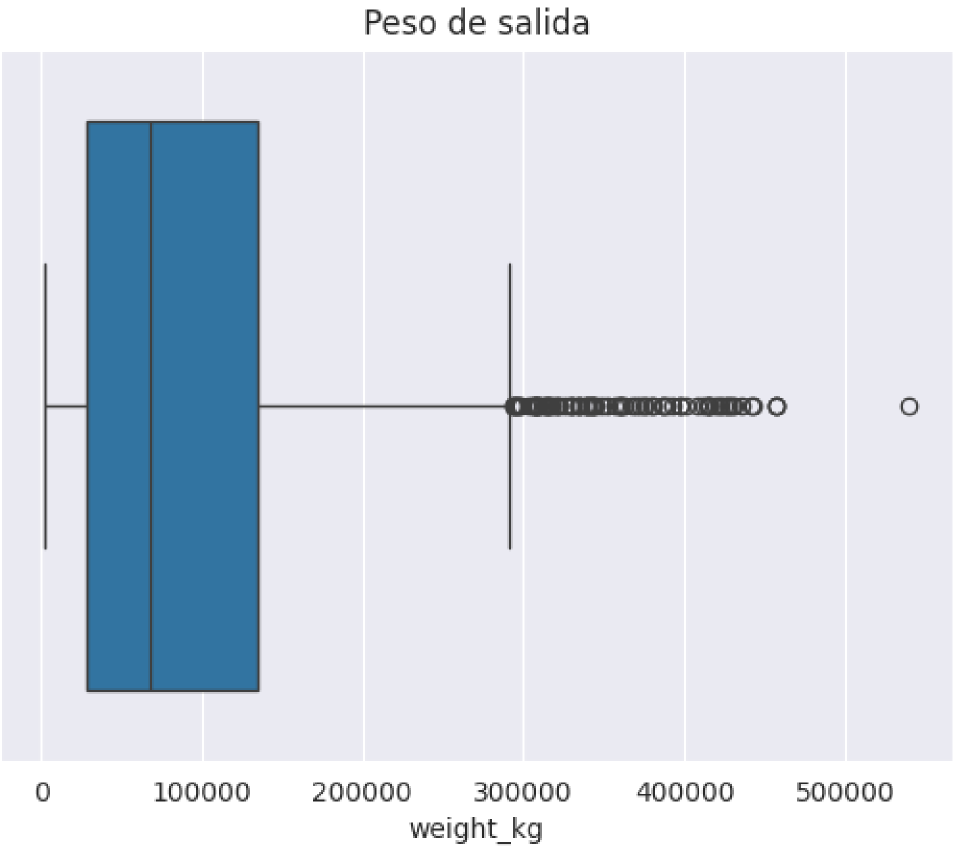

Peso de entrada
0.0
0.2
0.4
0.6
0.8
1.0
1.2
input_weight_kg
1e6

1.1.4 weight_kg

```
[54]: # Histograma
plt.figure(figsize=(18, 6))
do_histogram(most_used_paints_df, 'weight_kg', 80, title='Peso de salida')

# Boxplot
do_boxplot(most_used_paints_df, 'weight_kg', title='Peso de salida')
```

```
Numero de bins: 80 | Tamaño de cada bin: 6731.98
```

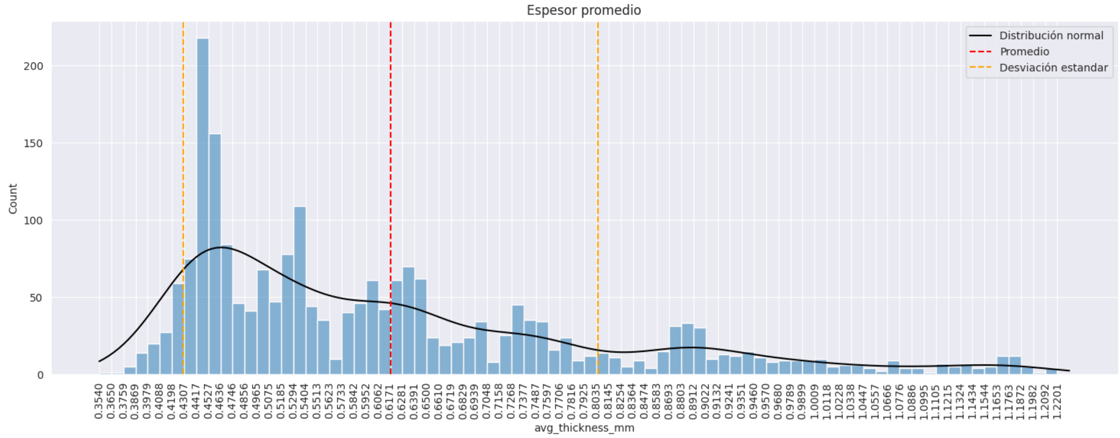


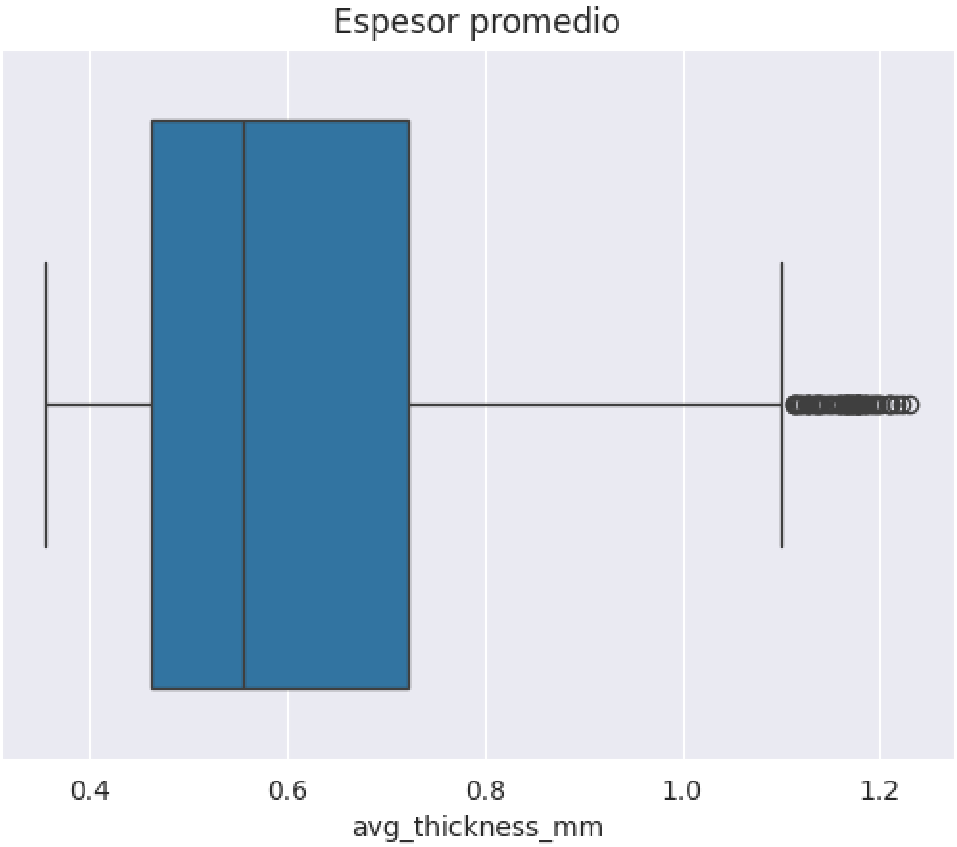


1.1.5 avg_thickness_mm

```
[55]: # Histograma
plt.figure(figsize=(18, 6))
do_histogram(most_used_paints_df, 'avg_thickness_mm', 80, title='Espesor␣
 ↪promedio')

# Boxplot
do_boxplot(most_used_paints_df, 'avg_thickness_mm', title='Espesor promedio')
```

```
Numero de bins: 80 | Tamaño de cada bin: 0.01
```

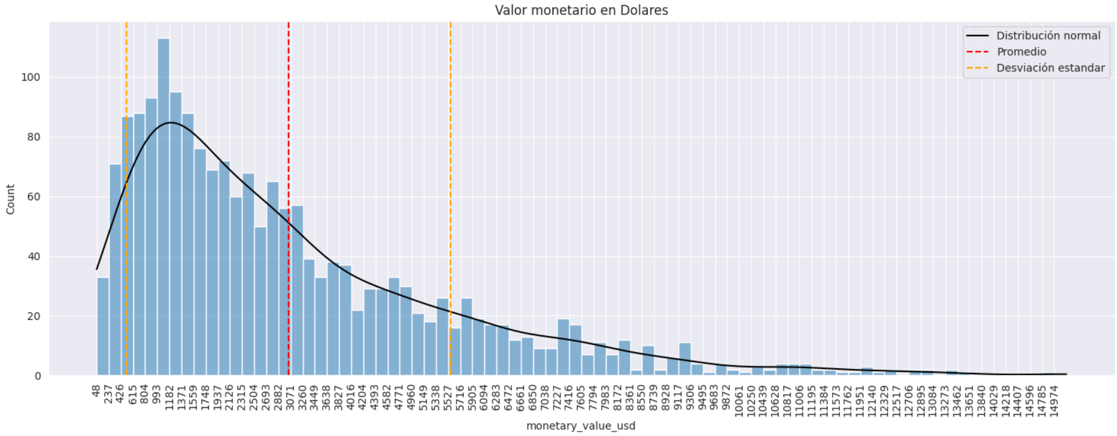

Espesor promedio
Distribución normal
Promedio
Desviación estandar
Count
0
50
100
150
200
0.3540
0.3650
0.3759
0.3869
0.3979
0.4088
0.4198
0.4307
0.4417
0.4527
0.4636
0.4746
0.4856
0.4965
0.5075
0.5185
0.5294
0.5404
0.5513
0.5623
0.5733
0.5842
0.5952
0.6062
0.6171
0.6281
0.6391
0.6500
0.6610
0.6719
0.6829
0.6939
0.7048
0.7158
0.7268
0.7377
0.7487
0.7597
0.7706
0.7816
0.7925
0.8035
0.8145
0.8254
0.8364
0.8474
0.8583
0.8693
0.8803
0.8912
0.9022
0.9132
0.9241
0.9351
0.9460
0.9570
0.9680
0.9789
0.9899
1.0009
1.0118
1.0228
1.0338
1.0447
1.0557
1.0666
1.0776
1.0886
1.0995
1.1105
1.1215
1.1324
1.1434
1.1544
1.1653
1.1763
1.1872
1.1982
1.2092
1.2201
avg_thickness_mm

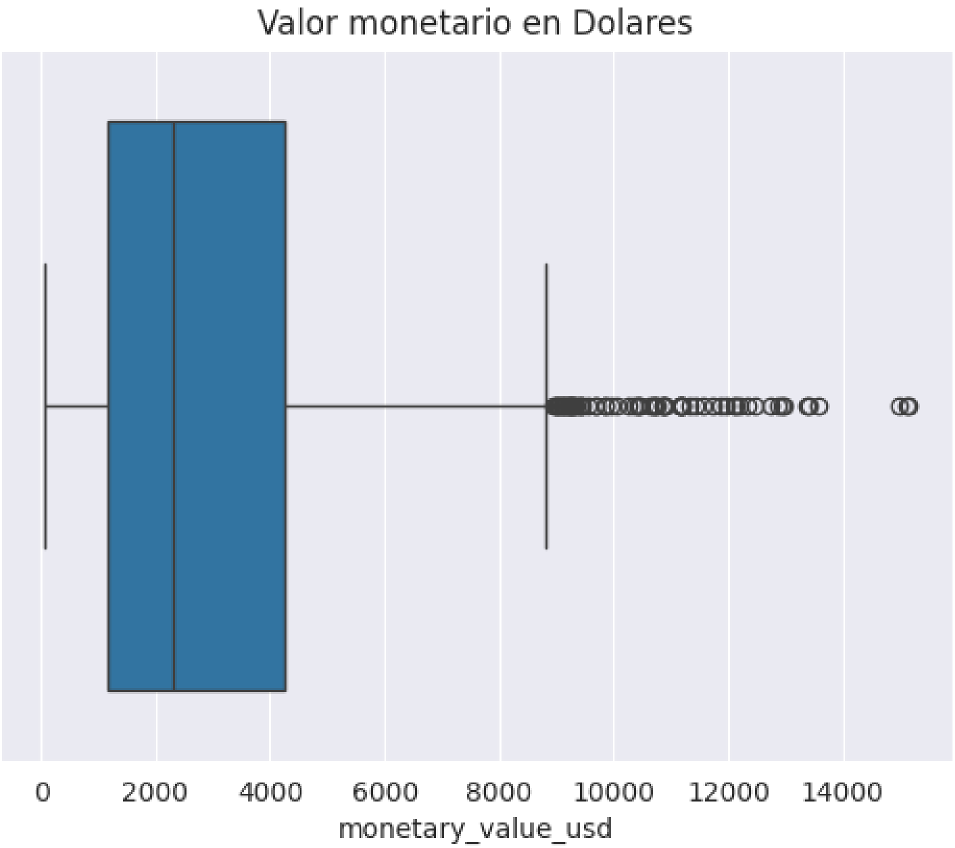

Espesor promedio
0.4
0.6
0.8
1.0
1.2
avg_thickness_mm

1.1.6 monetary_value_usd

[56]:
```
# Histograma
plt.figure(figsize=(18, 6))
do_histogram(most_used_paints_df, 'monetary_value_usd', 80, title='Valor␣
  ↪monetario en dólares')

# Boxplot
do_boxplot(most_used_paints_df, 'monetary_value_usd', title='Valor monetario en␣
  ↪dólares')
```

```
Numero de bins: 80 | Tamaño de cada bin: 188.93
```

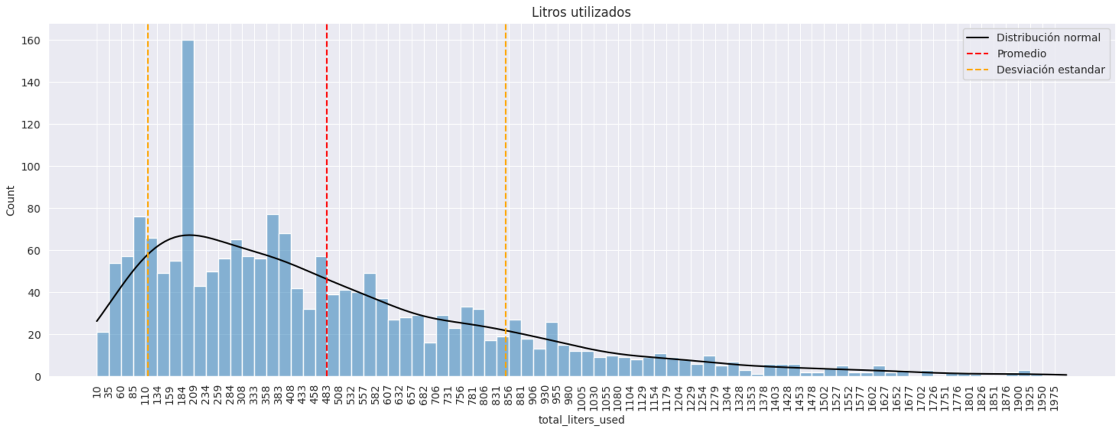


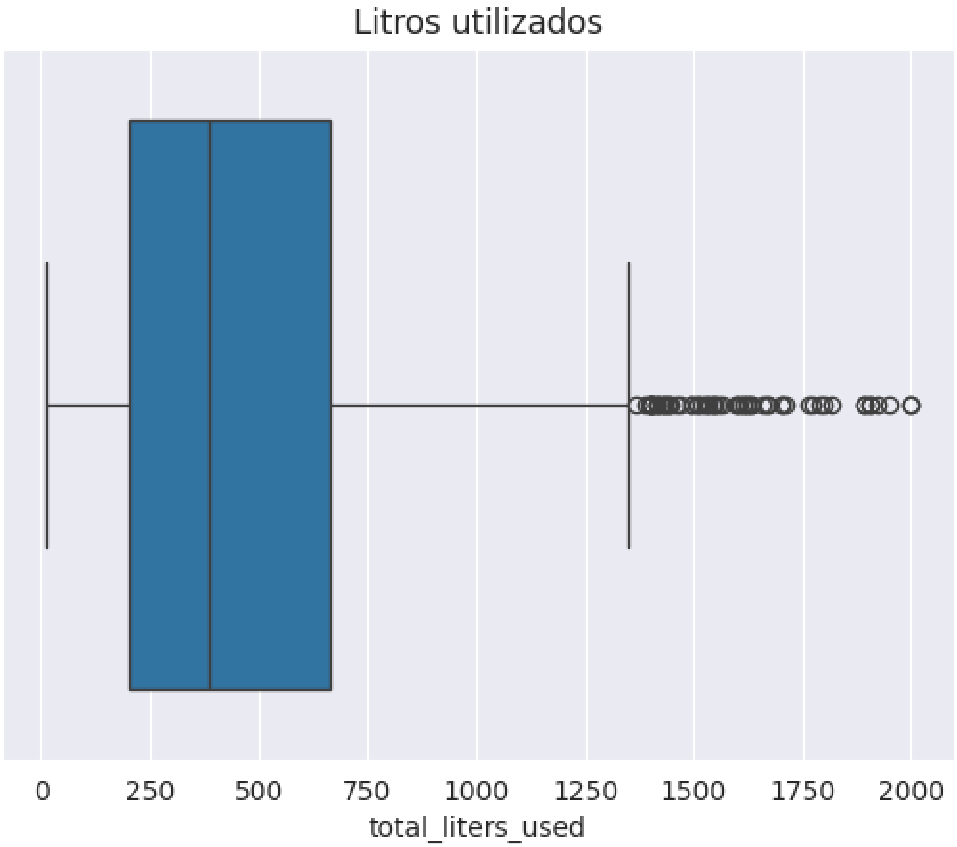


1.1.7 total_liters_used

[57]:
```
# Histograma
plt.figure(figsize=(18, 6))
do_histogram(most_used_paints_df, 'total_liters_used', 80, title='Litros␣
 ↪utilizados')

# Boxplot
do_boxplot(most_used_paints_df, 'total_liters_used', title='Litros utilizados')
```

```
Numero de bins: 80 | Tamaño de cada bin: 24.88
```

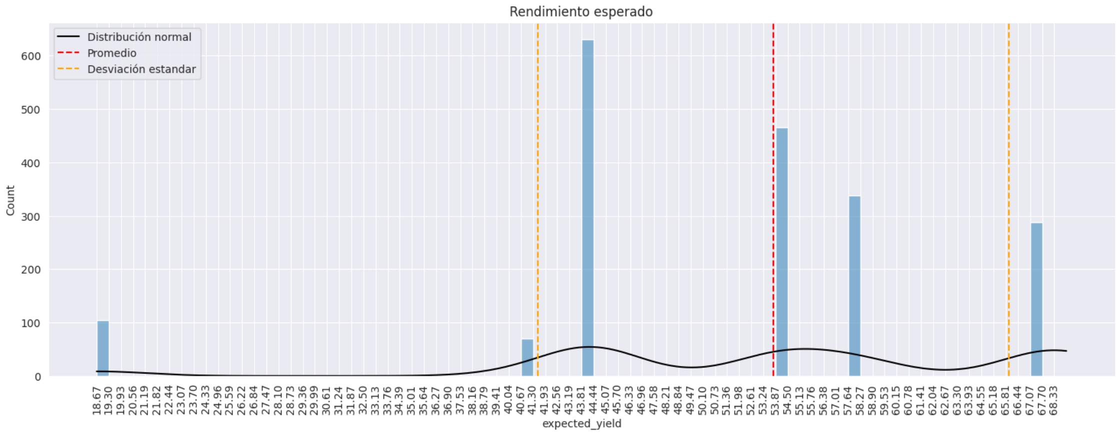

Litros utilizados
Distribución normal
Promedio
Desviación estandar
Count
0
20
40
60
80
100
120
140
160
10 35 60 85 110 134 159 184 209 234 259 284 308 333 358 383 408 433 458 483 508 532 557 582 607 632 657 682 706 731 756 781 806 831 856 881 906 930 955 980 1005 1030 1055 1080 1104 1129 1154 1179 1204 1229 1254 1279 1304 1328 1353 1378 1403 1428 1453 1478 1502 1527 1552 1577 1602 1627 1652 1677 1702 1726 1751 1776 1801 1826 1851 1876 1900 1925 1950 1975
total_liters_used

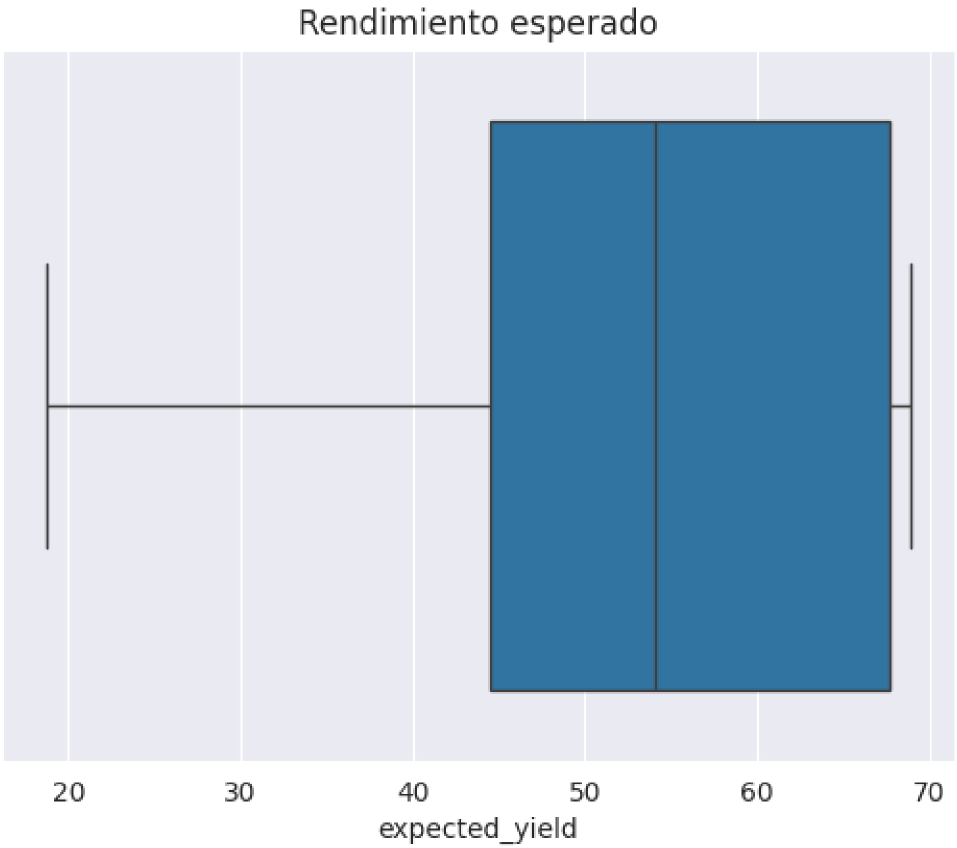

Litros utilizados
0
250
500
750
1000
1250
1500
1750
2000
total_liters_used

1.1.8 expected_yield

[58]:
```python
# Histograma
plt.figure(figsize=(18, 6))
do_histogram(most_used_paints_df, 'expected_yield', 80, title='Rendimiento␣
↪esperado')

# Boxplot
do_boxplot(most_used_paints_df, 'expected_yield', title='Rendimiento esperado')
```

```
Numero de bins: 80 | Tamaño de cada bin: 0.63
```

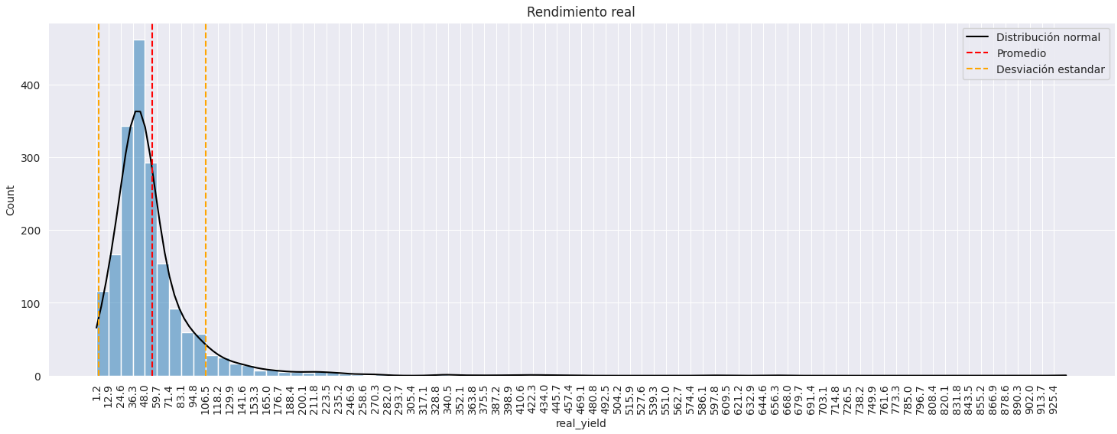


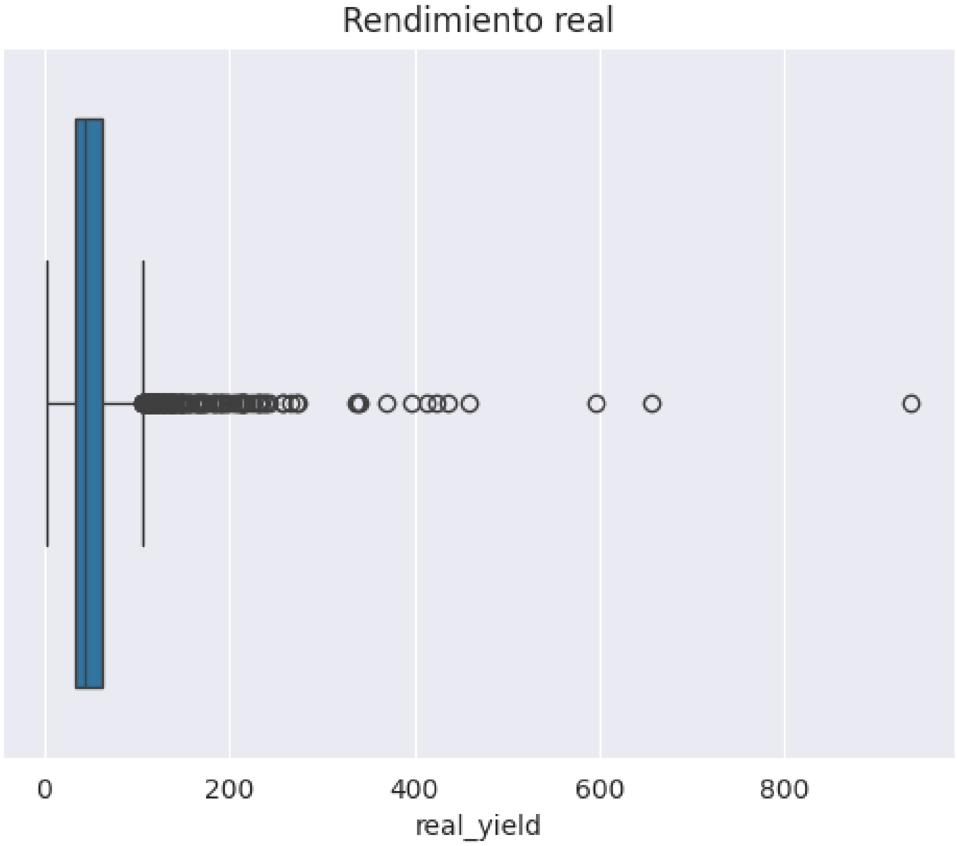


1.1.9 real_yield

```
[59]: # Histograma
plt.figure(figsize=(18, 6))
do_histogram(most_used_paints_df, 'real_yield', 80, title='Rendimiento real')

# Boxplot
do_boxplot(most_used_paints_df, 'real_yield', title='Rendimiento real')
```

```
Numero de bins: 80 | Tamaño de cada bin: 11.70
```

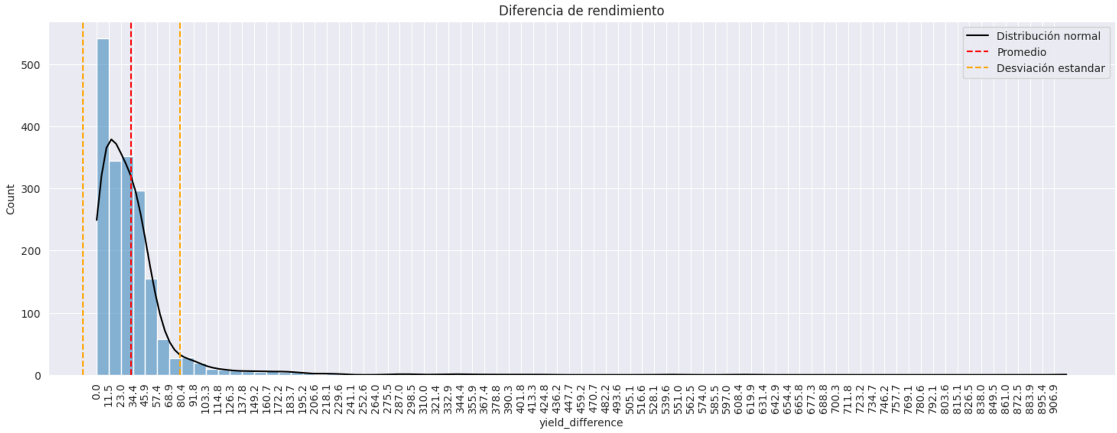

Rendimiento real
Distribución normal
Promedio
Desviación estandar
Count
400
300
200
100
0
1.2
12.9
24.6
36.3
48.0
59.7
71.4
83.1
94.8
106.5
118.2
129.9
141.6
153.3
165.0
176.7
188.4
200.1
211.8
223.5
235.2
246.9
258.6
270.3
282.0
293.7
305.4
317.1
328.8
340.5
352.1
363.8
375.5
387.2
398.9
410.6
422.3
434.0
445.7
457.4
469.1
480.8
492.5
504.2
515.9
527.6
539.3
551.0
562.7
574.4
586.1
597.8
609.5
621.2
632.9
644.6
656.3
668.0
679.7
691.4
703.1
714.8
726.5
738.2
749.9
761.6
773.3
785.0
796.7
808.4
820.1
831.8
843.5
855.2
866.9
878.6
890.3
902.0
913.7
925.4
real_yield

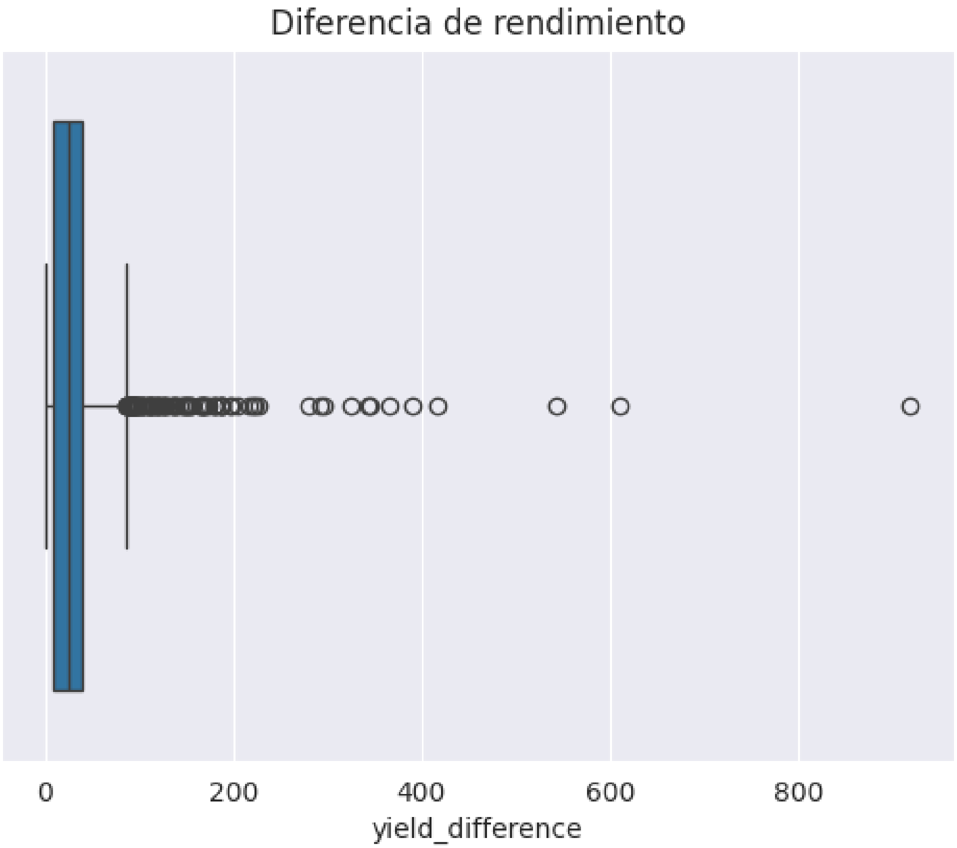

Rendimiento real
0
200
400
600
800
real_yield

1.1.10 yield_difference

```
[60]: # Histograma
      plt.figure(figsize=(18, 6))
      do_histogram(most_used_paints_df, 'yield_difference', 80, title='Diferencia de␣
        ↪rendimiento')

      # Boxplot
      do_boxplot(most_used_paints_df, 'yield_difference', title='Diferencia de␣
        ↪rendimiento')
```

```
Numero de bins: 80 | Tamaño de cada bin: 11.48
```





1.2 Variables cualitativas

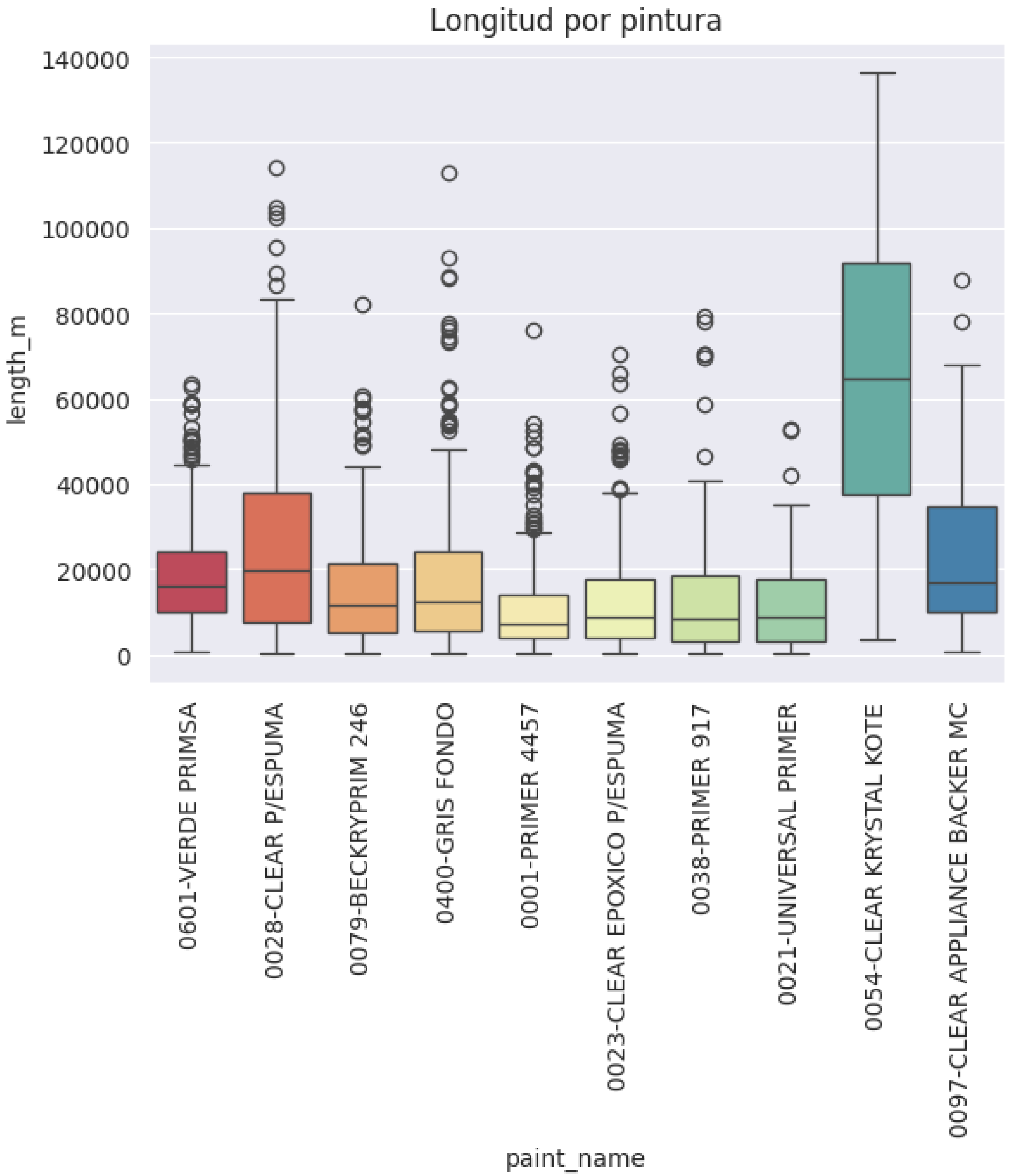
1.2.1 paint_name

[61]:

```
described_df = most_used_paints_df.groupby('paint_name').describe().
 ↪stack(level=0, future_stack=True)[
    ['75%', '50%', '25%', 'min', 'max']]
described_df = described_df.assign(
    IQR=lambda x: x['75%'] - x['25%'],
    top_whisker=lambda x: (x['75%'] + 1.5 * x['IQR']).combine(x['max'], min),
    bottom_whisker=lambda x: (x['25%'] - 1.5 * x['IQR']).combine(x['min'], max)
)[['bottom_whisker', '25%', '50%', '75%', 'top_whisker', 'IQR']].stack().
 ↪unstack(level=-2).unstack(level=-1)
```

[62]:

```
do_boxplot(most_used_paints_df, x='paint_name', y='length_m', rotate=True,␣
 ↪title='Longitud por pintura')
described_df.loc[:, ['length_m']]
```



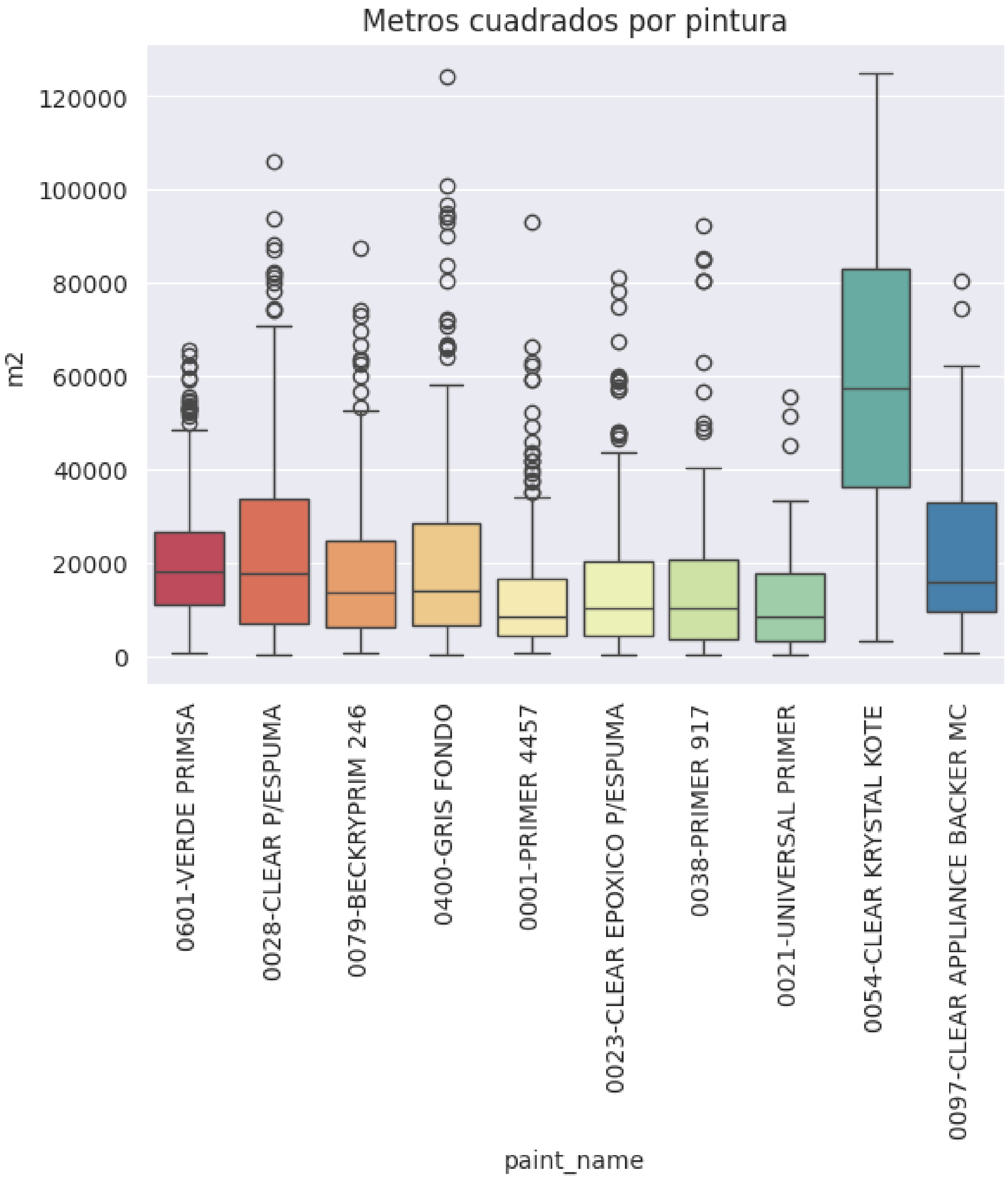
```
[62]:                                    length_m                                \
                                   bottom_whisker      25%      50%       75%
      paint_name
      0001-PRIMER 4457                      426.0  3878.25   7335.5  13925.75
      0021-UNIVERSAL PRIMER                 435.0  3313.00   8758.0  17884.00
      0023-CLEAR EPOXICO P/ESPUMA           234.0  3935.00   8735.0  17517.00
      0028-CLEAR P/ESPUMA                   318.0  7448.25  19832.0  37926.25
      0038-PRIMER 917                       311.0  3304.25   8426.5  18561.25
```

```
0054-CLEAR KRYSTAL KOTE            3459.0  37600.50  64600.5  91866.00
0079-BECKRYPRIM 246                 417.0   5172.75  11449.5  21311.50
0097-CLEAR APPLIANCE BACKER MC      513.0  10211.00  17095.5  34850.75
0400-GRIS FONDO                     404.0   5740.00  12406.0  24378.00
0601-VERDE PRIMSA                   568.0  10049.25  15988.0  24117.50
```

```
                               top_whisker       IQR
paint_name
0001-PRIMER 4457                 28997.000  10047.50
0021-UNIVERSAL PRIMER            39740.500  14571.00
0023-CLEAR EPOXICO P/ESPUMA      37890.000  13582.00
0028-CLEAR P/ESPUMA              83643.250  30478.00
0038-PRIMER 917                  41446.750  15257.00
0054-CLEAR KRYSTAL KOTE         136332.000  54265.50
0079-BECKRYPRIM 246              45519.625  16138.75
0097-CLEAR APPLIANCE BACKER MC   71810.375  24639.75
0400-GRIS FONDO                  52335.000  18638.00
0601-VERDE PRIMSA                45219.875  14068.25
```

[63]:

```
do_boxplot(most_used_paints_df, x='paint_name', y='m2', rotate=True,␣
 ↪title='Metros cuadrados por pintura')
described_df.loc[:, ['m2']]
```

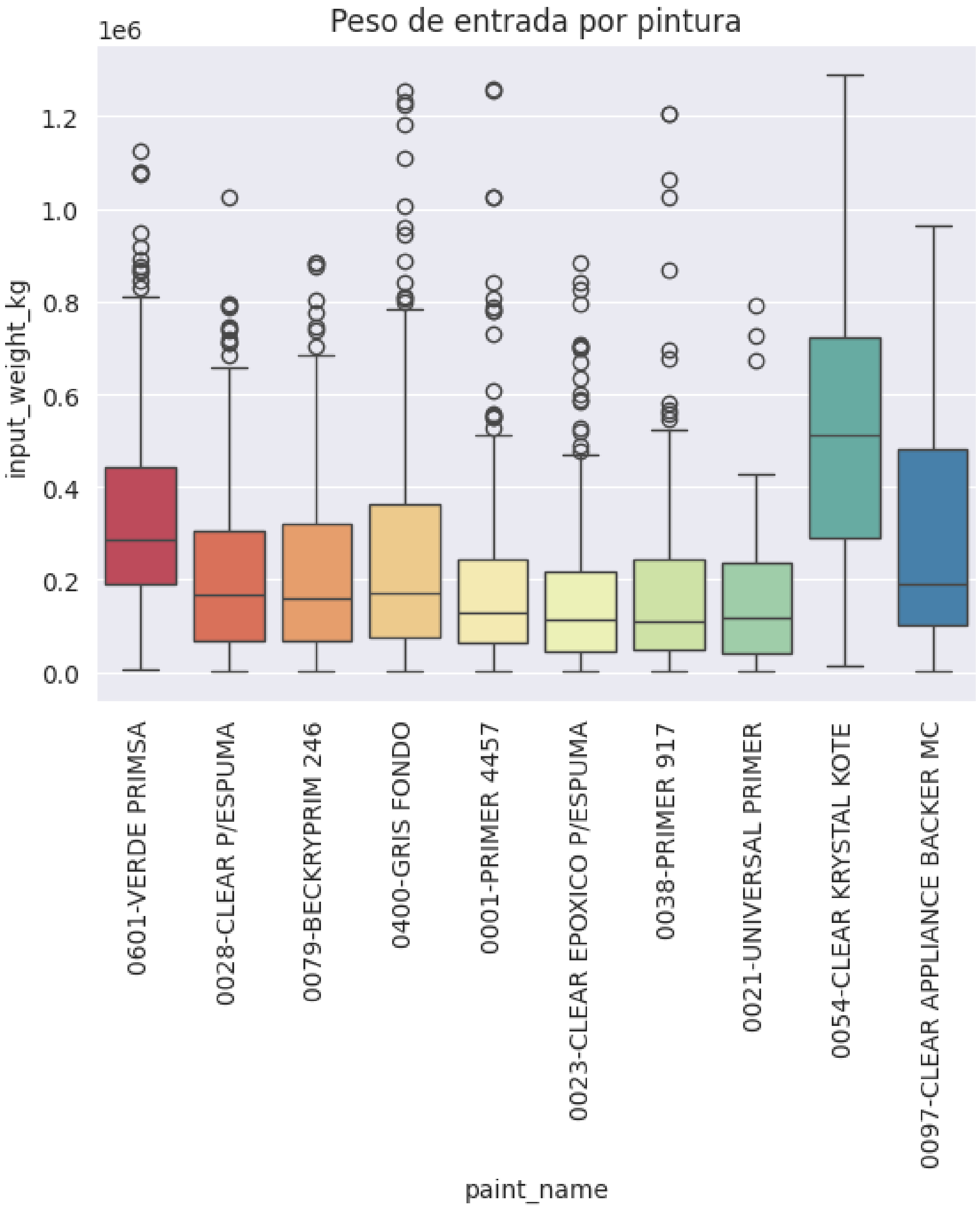


```
[63]:                                    m2                              \
                              bottom_whisker          25%          50%
      paint_name
      0001-PRIMER 4457               522.276  4212.587448  8256.402860
      0021-UNIVERSAL PRIMER          410.640  3291.664000  8450.560000
      0023-CLEAR EPOXICO P/ESPUMA    214.812  4481.342000 10434.515000
      0028-CLEAR P/ESPUMA            389.232  6950.852250 17541.870500
      0038-PRIMER 917                384.085  3622.422500 10158.390005
```

```
0054-CLEAR KRYSTAL KOTE               3130.395  36156.072525  57245.597913
0079-BECKRYPRIM 246                    508.740   6097.863250  13688.266500
0097-CLEAR APPLIANCE BACKER MC         469.395   9361.873250  15671.320500
0400-GRIS FONDO                        447.632   6524.412608  14037.595000
0601-VERDE PRIMSA                      679.896  11133.083750  18007.752355

                                         75%    top_whisker           IQR
paint_name
0001-PRIMER 4457                16409.631722   34705.198135  12197.044275
0021-UNIVERSAL PRIMER           17854.161000   39697.906500  14562.497000
0023-CLEAR EPOXICO P/ESPUMA     20326.523230   44094.295075  15845.181230
0028-CLEAR P/ESPUMA             33613.029000   73606.294125  26662.176750
0038-PRIMER 917                 20685.816500   46280.907500  17063.394000
0054-CLEAR KRYSTAL KOTE         82826.038707  124782.771000  46669.966182
0079-BECKRYPRIM 246             24702.377750   52609.149500  18604.514500
0097-CLEAR APPLIANCE BACKER MC  33047.438765   68575.787037  23685.565515
0400-GRIS FONDO                 28401.492000   61217.111087  21877.079392
0601-VERDE PRIMSA               26450.989275   49427.847562  15317.905525
```

[64]:

```
do_boxplot(most_used_paints_df, x='paint_name', y='input_weight_kg', rotate=True, title='Peso de entrada por pintura')
described_df.loc[:, ['input_weight_kg']]
```



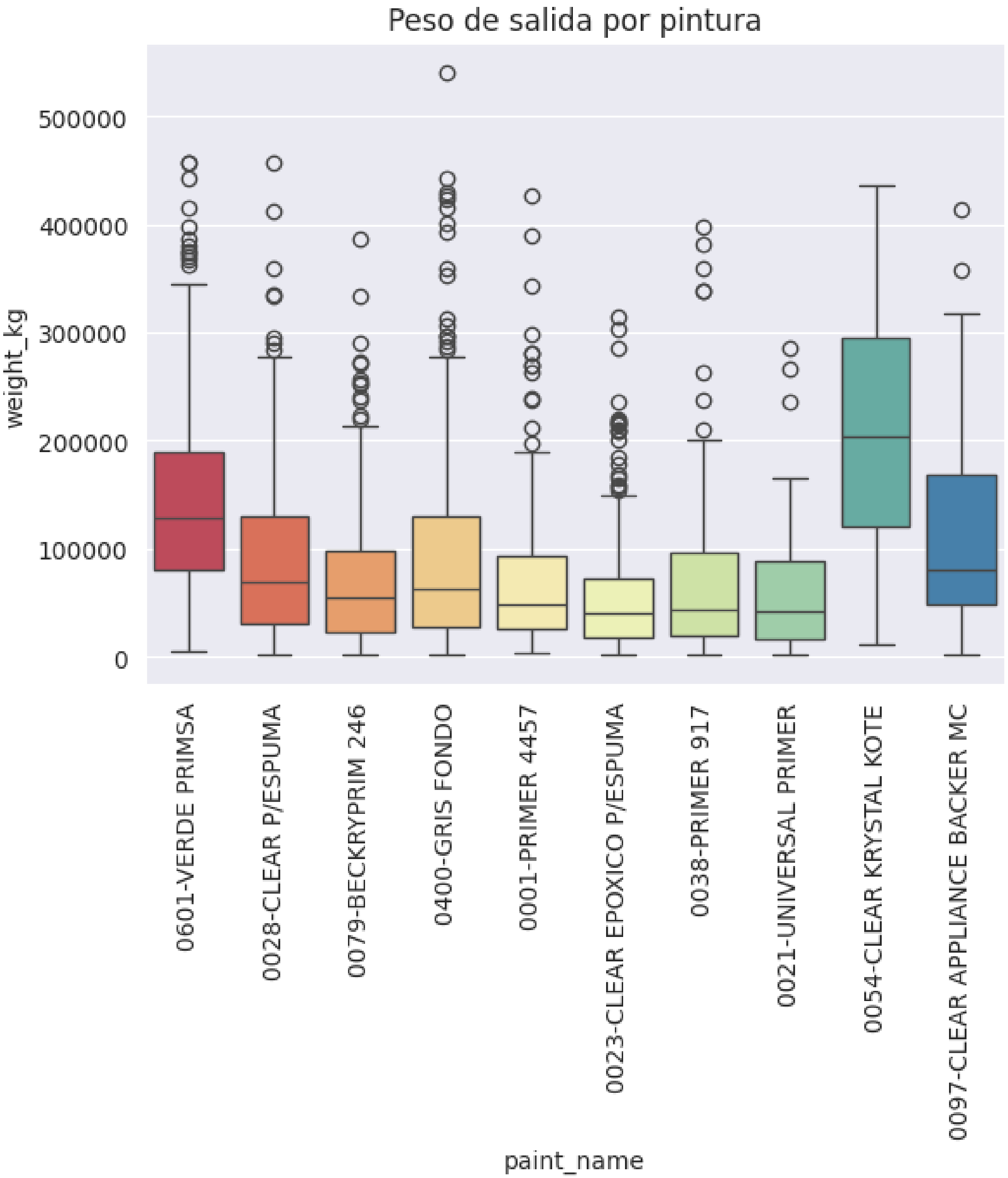
[64]:

```
                               input_weight_kg                       \
                                bottom_whisker       25%       50%
paint_name
0001-PRIMER 4457                        3330.0  63243.50  128772.5
0021-UNIVERSAL PRIMER                   2103.0  40042.00  117581.0
0023-CLEAR EPOXICO P/ESPUMA             2010.0  43581.50  113139.0
0028-CLEAR P/ESPUMA                     2245.0  68103.75  167603.0
```

```
0038-PRIMER 917                         3290.0   49072.50  111397.0
0054-CLEAR KRYSTAL KOTE                15852.0  291702.00  510880.5
0079-BECKRYPRIM 246                     2700.0   66131.50  158927.5
0097-CLEAR APPLIANCE BACKER MC          2435.0  101044.50  188574.0
0400-GRIS FONDO                         2590.0   76020.00  170821.0
0601-VERDE PRIMSA                       4705.0  191139.00  287074.0


                                      75%  top_whisker        IQR
paint_name
0001-PRIMER 4457                243736.25   514475.375  180492.75
0021-UNIVERSAL PRIMER           235099.00   527684.500  195057.00
0023-CLEAR EPOXICO P/ESPUMA     215903.00   474385.250  172321.50
0028-CLEAR P/ESPUMA             304719.00   659641.875  236615.25
0038-PRIMER 917                 242688.25   533111.875  193615.75
0054-CLEAR KRYSTAL KOTE         723744.00  1289409.000  432042.00
0079-BECKRYPRIM 246             319745.75   700167.125  253614.25
0097-CLEAR APPLIANCE BACKER MC  480606.25   966132.000  379561.75
0400-GRIS FONDO                 361518.00   789765.000  285498.00
0601-VERDE PRIMSA               441775.25   817729.625  250636.25
```

[65]:

```
do_boxplot(most_used_paints_df, x='paint_name', y='weight_kg', rotate=True,␣
 ↪title='Peso de salida por pintura')
described_df.loc[:, ['weight_kg']]
```



[65]:

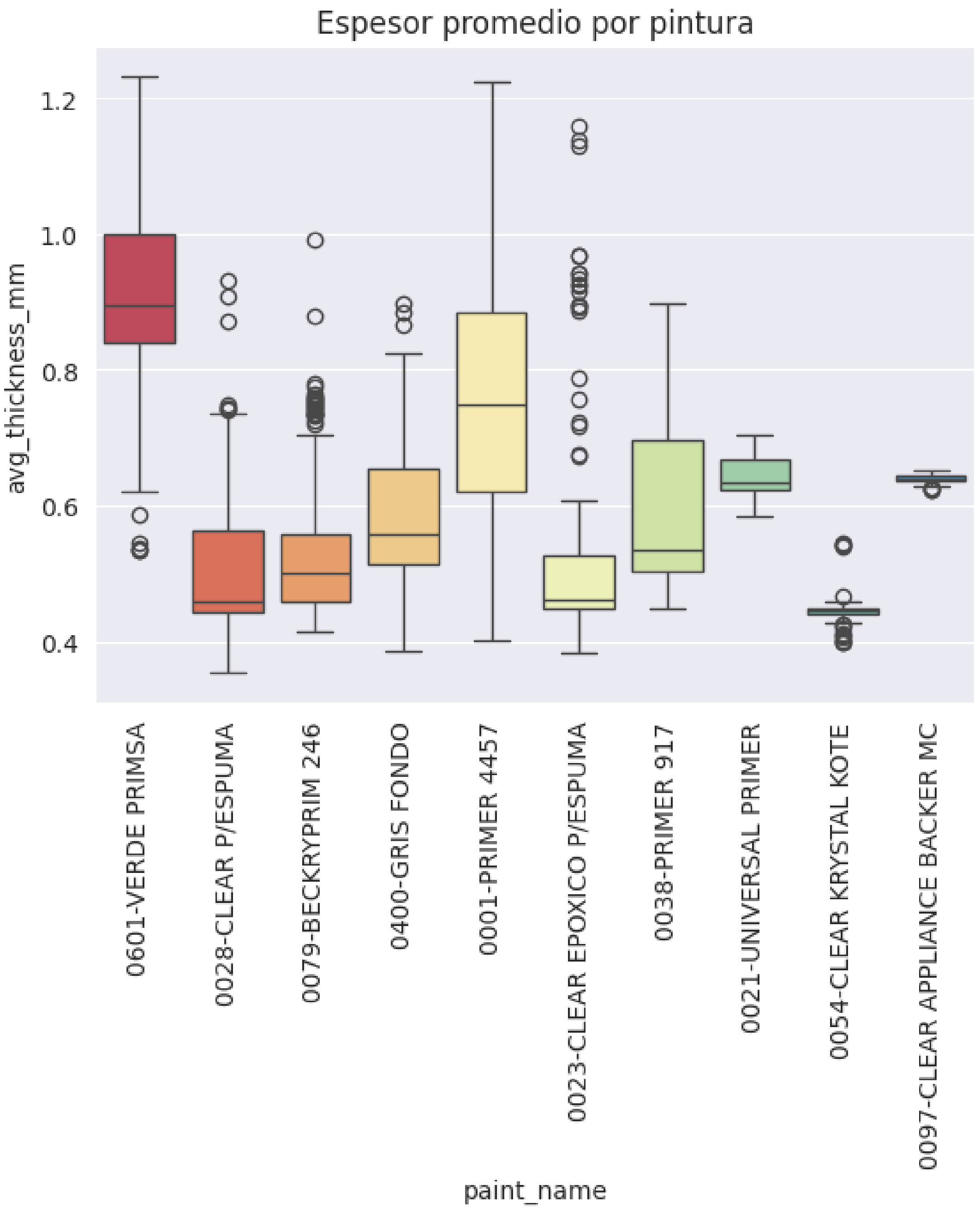
```
                                 weight_kg                                 \
                            bottom_whisker        25%       50%        75%
paint_name
0001-PRIMER 4457                    3170.0   25969.00   48088.0   93281.00
0021-UNIVERSAL PRIMER               2093.0   16084.00   42266.0   89158.00
0023-CLEAR EPOXICO P/ESPUMA         1938.0   17671.50   40398.0   72369.00
0028-CLEAR P/ESPUMA                 1630.0   30382.25   69950.5  129835.50
0038-PRIMER 917                     2284.0   20247.25   43784.0   96304.00
```

```
0054-CLEAR KRYSTAL KOTE            11037.0   120700.50  203992.5  294857.25
0079-BECKRYPRIM 246                 2672.0    21982.50   55569.5   98803.00
0097-CLEAR APPLIANCE BACKER MC      2359.0    47637.50   80404.0  169326.00
0400-GRIS FONDO                     1800.0    28295.00   63385.0  130082.00
0601-VERDE PRIMSA                   4675.0    80783.75  128362.0  189371.25
```

```
                               top_whisker        IQR
paint_name
0001-PRIMER 4457                194249.000   67312.00
0021-UNIVERSAL PRIMER           198769.000   73074.00
0023-CLEAR EPOXICO P/ESPUMA     154415.250   54697.50
0028-CLEAR P/ESPUMA             279015.375   99453.25
0038-PRIMER 917                 210389.125   76056.75
0054-CLEAR KRYSTAL KOTE         435552.000  174156.75
0079-BECKRYPRIM 246             214033.750   76820.50
0097-CLEAR APPLIANCE BACKER MC  351858.750  121688.50
0400-GRIS FONDO                 282762.500  101787.00
0601-VERDE PRIMSA               352252.500  108587.50
```

[66]:

```
do_boxplot(most_used_paints_df, x='paint_name', y='avg_thickness_mm',␣
 ↪rotate=True, title='Espesor promedio por pintura')
described_df.loc[:, ['avg_thickness_mm']]
```



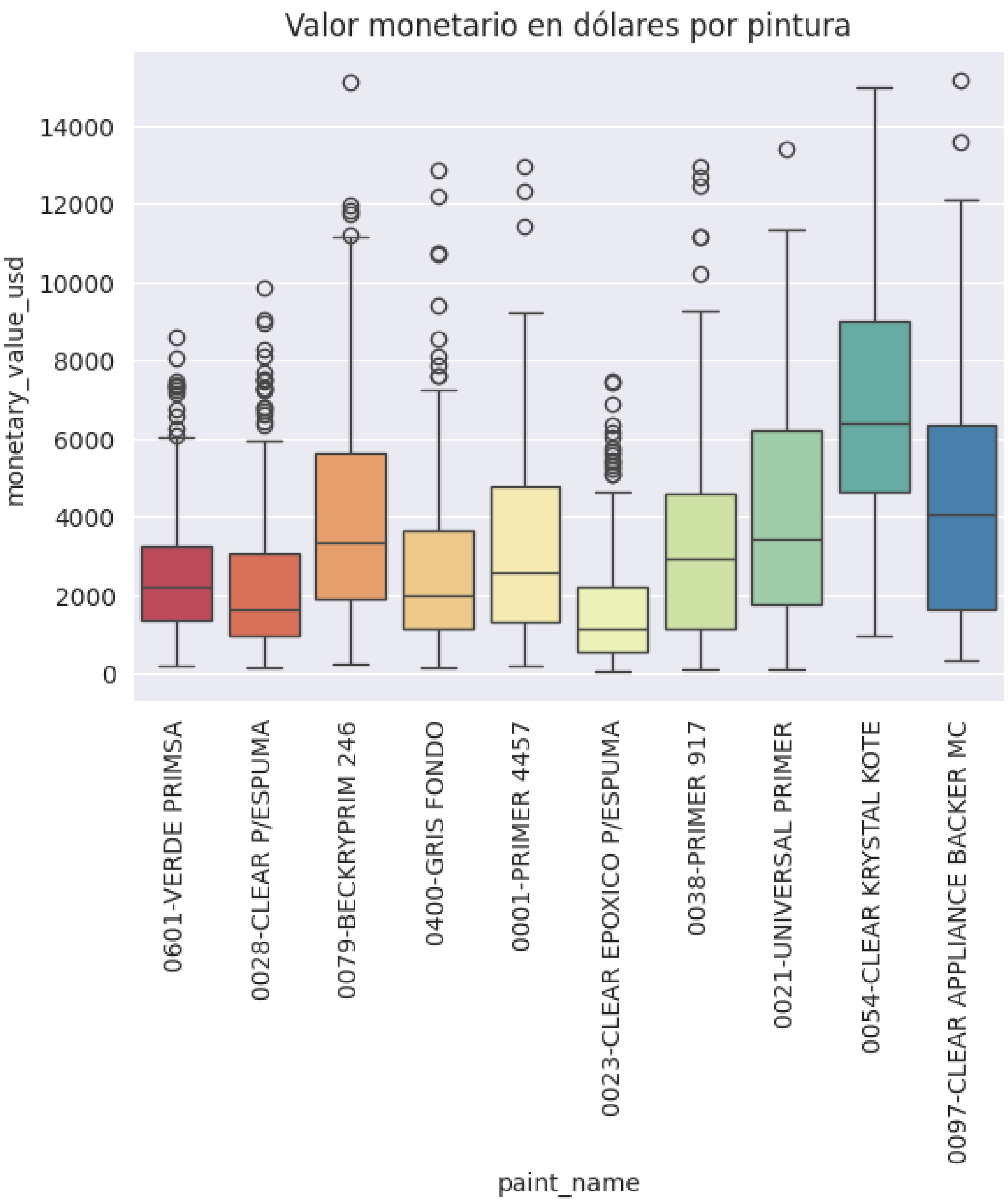
[66]:

```
                               avg_thickness_mm                                \
                                 bottom_whisker       25%       50%       75%
paint_name
0001-PRIMER 4457                       0.401767  0.620437  0.747336  0.885434
0021-UNIVERSAL PRIMER                  0.583000  0.622157  0.633300  0.667077
0023-CLEAR EPOXICO P/ESPUMA            0.383250  0.448664  0.462000  0.525622
0028-CLEAR P/ESPUMA                    0.354000  0.443937  0.459192  0.562927
```

```
0038-PRIMER 917                       0.449477  0.503714  0.534188  0.696426
0054-CLEAR KRYSTAL KOTE               0.427580  0.440724  0.445426  0.449486
0079-BECKRYPRIM 246                   0.415000  0.457905  0.500586  0.559317
0097-CLEAR APPLIANCE BACKER MC        0.627290  0.636647  0.639583  0.642886
0400-GRIS FONDO                       0.386750  0.514300  0.558300  0.654939
0601-VERDE PRIMSA                     0.602511  0.840483  0.895586  0.999131


                               top_whisker       IQR
paint_name
0001-PRIMER 4457                  1.223000  0.264996
0021-UNIVERSAL PRIMER             0.704000  0.044920
0023-CLEAR EPOXICO P/ESPUMA       0.641059  0.076958
0028-CLEAR P/ESPUMA               0.741412  0.118990
0038-PRIMER 917                   0.897000  0.192712
0054-CLEAR KRYSTAL KOTE           0.462630  0.008762
0079-BECKRYPRIM 246               0.711434  0.101411
0097-CLEAR APPLIANCE BACKER MC    0.651971  0.006238
0400-GRIS FONDO                   0.865898  0.140639
0601-VERDE PRIMSA                 1.231100  0.158648
```

[67]:

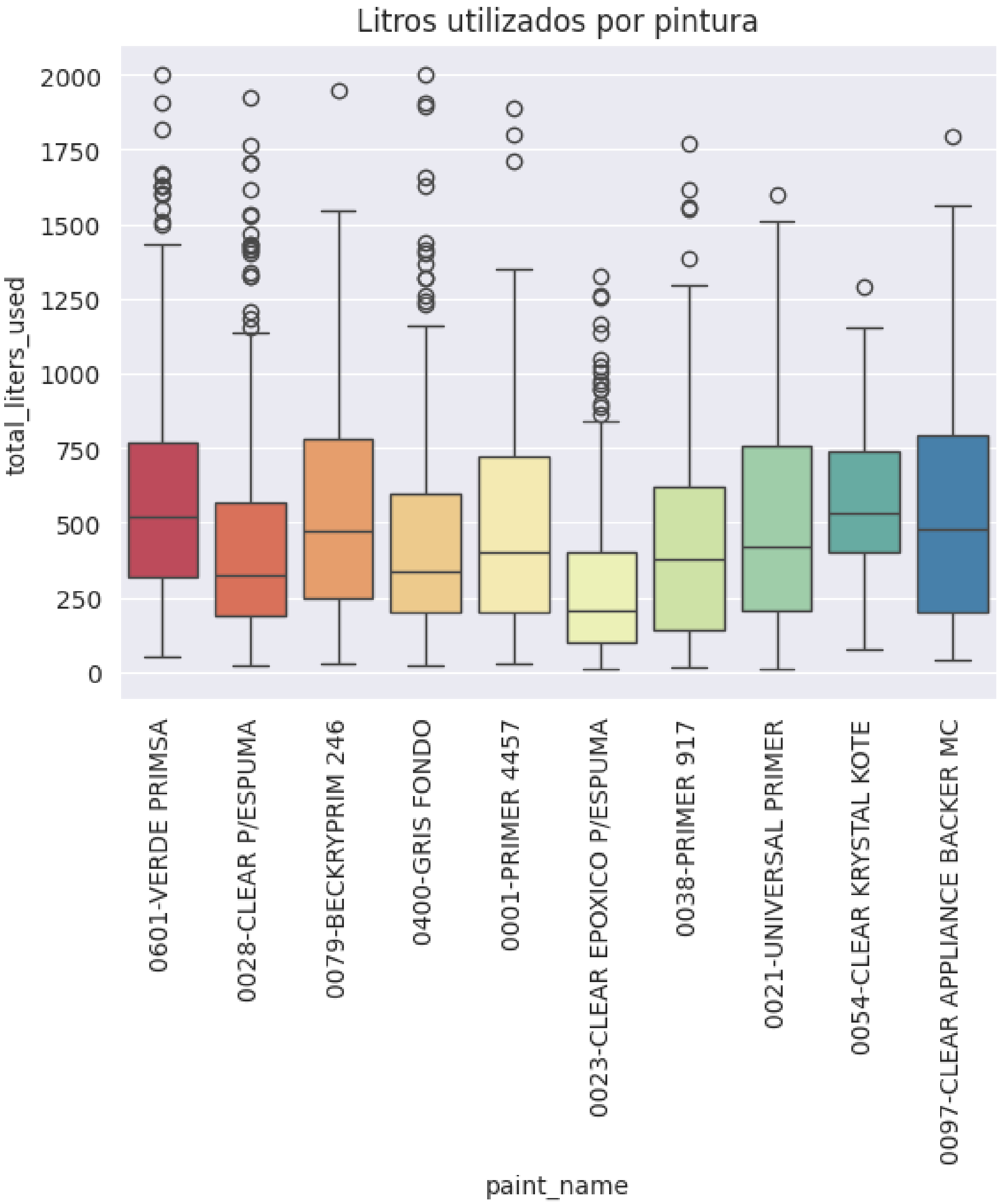
```
do_boxplot(most_used_paints_df, x='paint_name', y='monetary_value_usd',␣
 ↪rotate=True,
           title='Valor monetario en dólares por pintura')
described_df.loc[:, ['total_liters_used']]
```



```
[67]:                                  total_liters_used                               \
                                           bottom_whisker       25%      50%       75%
      paint_name
      0001-PRIMER 4457                             30.000  200.0000  400.000  720.0000
      0021-UNIVERSAL PRIMER                        10.000  208.5000  417.500  756.7500
      0023-CLEAR EPOXICO P/ESPUMA                  10.000  100.1875  207.375  403.8125
      0028-CLEAR P/ESPUMA                          25.000  189.2500  322.250  567.7500
```

```
0038-PRIMER 917                         15.000  143.0000  378.500  620.6250
0054-CLEAR KRYSTAL KOTE                 75.000  400.0000  535.000  737.5000
0079-BECKRYPRIM 246                     30.000  250.0000  470.000  783.7500
0097-CLEAR APPLIANCE BACKER MC          39.787  199.2500  477.750  792.0000
0400-GRIS FONDO                         25.000  200.0000  335.000  600.0000
0601-VERDE PRIMSA                       50.000  320.0000  520.000  771.2500


                               top_whisker      IQR
paint_name
0001-PRIMER 4457                 1500.0000  520.000
0021-UNIVERSAL PRIMER            1579.1250  548.250
0023-CLEAR EPOXICO P/ESPUMA       859.2500  303.625
0028-CLEAR P/ESPUMA              1135.5000  378.500
0038-PRIMER 917                  1337.0625  477.625
0054-CLEAR KRYSTAL KOTE          1243.7500  337.500
0079-BECKRYPRIM 246              1584.3750  533.750
0097-CLEAR APPLIANCE BACKER MC   1681.1250  592.750
0400-GRIS FONDO                  1200.0000  400.000
0601-VERDE PRIMSA                1448.1250  451.250
```

[68]:
```
do_boxplot(most_used_paints_df, x='paint_name', y='total_liters_used',␣
 ↪rotate=True,
           title='Litros utilizados por pintura')
described_df.loc[:, ['total_liters_used']]
```

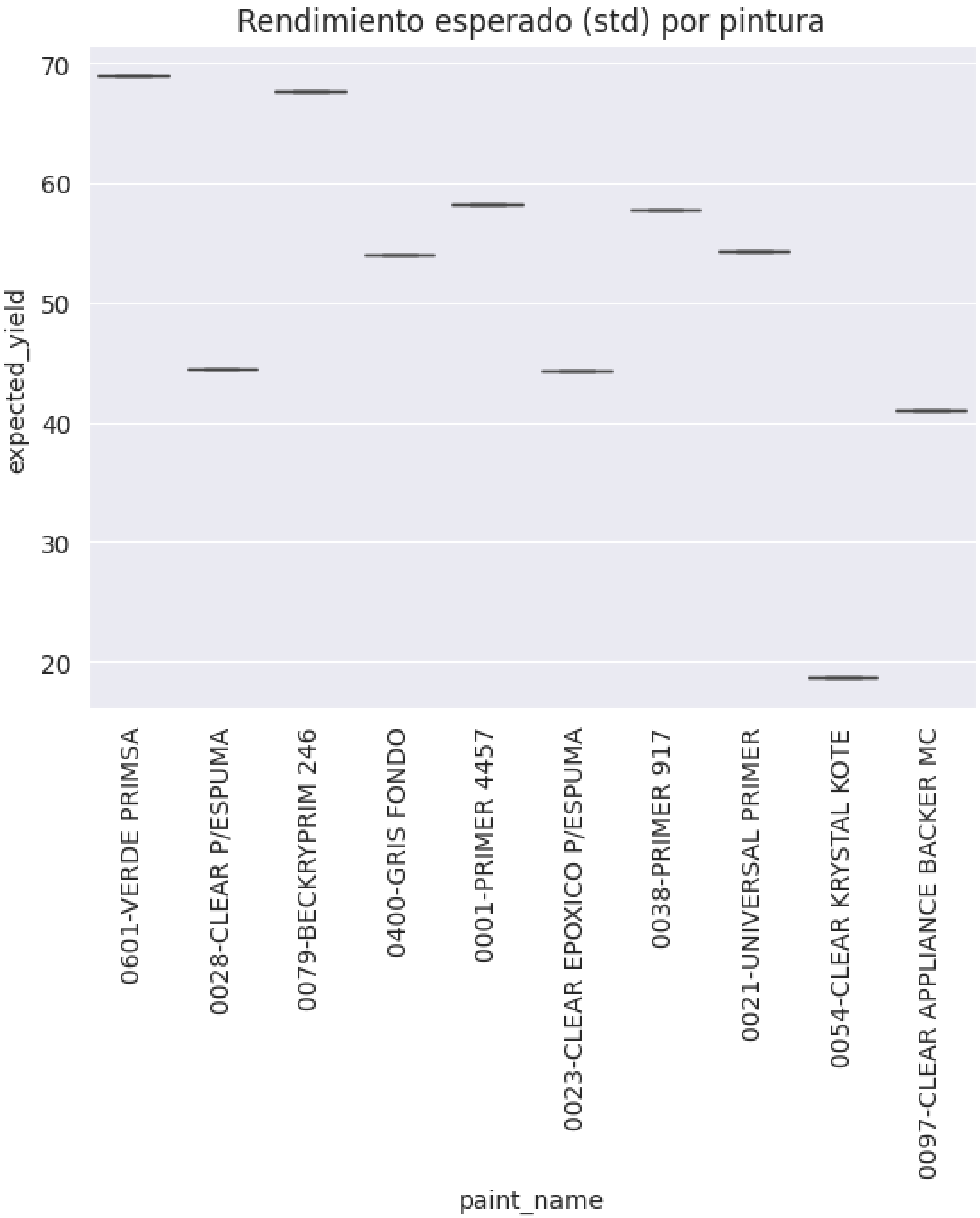


```
[68]:                             total_liters_used                                \
                                     bottom_whisker       25%      50%       75%
      paint_name
      0001-PRIMER 4457                       30.000  200.0000  400.000  720.0000
      0021-UNIVERSAL PRIMER                  10.000  208.5000  417.500  756.7500
      0023-CLEAR EPOXICO P/ESPUMA            10.000  100.1875  207.375  403.8125
      0028-CLEAR P/ESPUMA                    25.000  189.2500  322.250  567.7500
```

```
0038-PRIMER 917                          15.000  143.0000  378.500  620.6250
0054-CLEAR KRYSTAL KOTE                  75.000  400.0000  535.000  737.5000
0079-BECKRYPRIM 246                      30.000  250.0000  470.000  783.7500
0097-CLEAR APPLIANCE BACKER MC           39.787  199.2500  477.750  792.0000
0400-GRIS FONDO                          25.000  200.0000  335.000  600.0000
0601-VERDE PRIMSA                        50.000  320.0000  520.000  771.2500


                               top_whisker      IQR
paint_name
0001-PRIMER 4457                 1500.0000  520.000
0021-UNIVERSAL PRIMER            1579.1250  548.250
0023-CLEAR EPOXICO P/ESPUMA       859.2500  303.625
0028-CLEAR P/ESPUMA              1135.5000  378.500
0038-PRIMER 917                  1337.0625  477.625
0054-CLEAR KRYSTAL KOTE          1243.7500  337.500
0079-BECKRYPRIM 246              1584.3750  533.750
0097-CLEAR APPLIANCE BACKER MC   1681.1250  592.750
0400-GRIS FONDO                  1200.0000  400.000
0601-VERDE PRIMSA                1448.1250  451.250
```

[69]:

```python
do_boxplot(most_used_paints_df, x='paint_name', y='expected_yield', rotate=True,
           title='Rendimiento esperado (std) por pintura')
described_df.loc[:, ['expected_yield']]
```



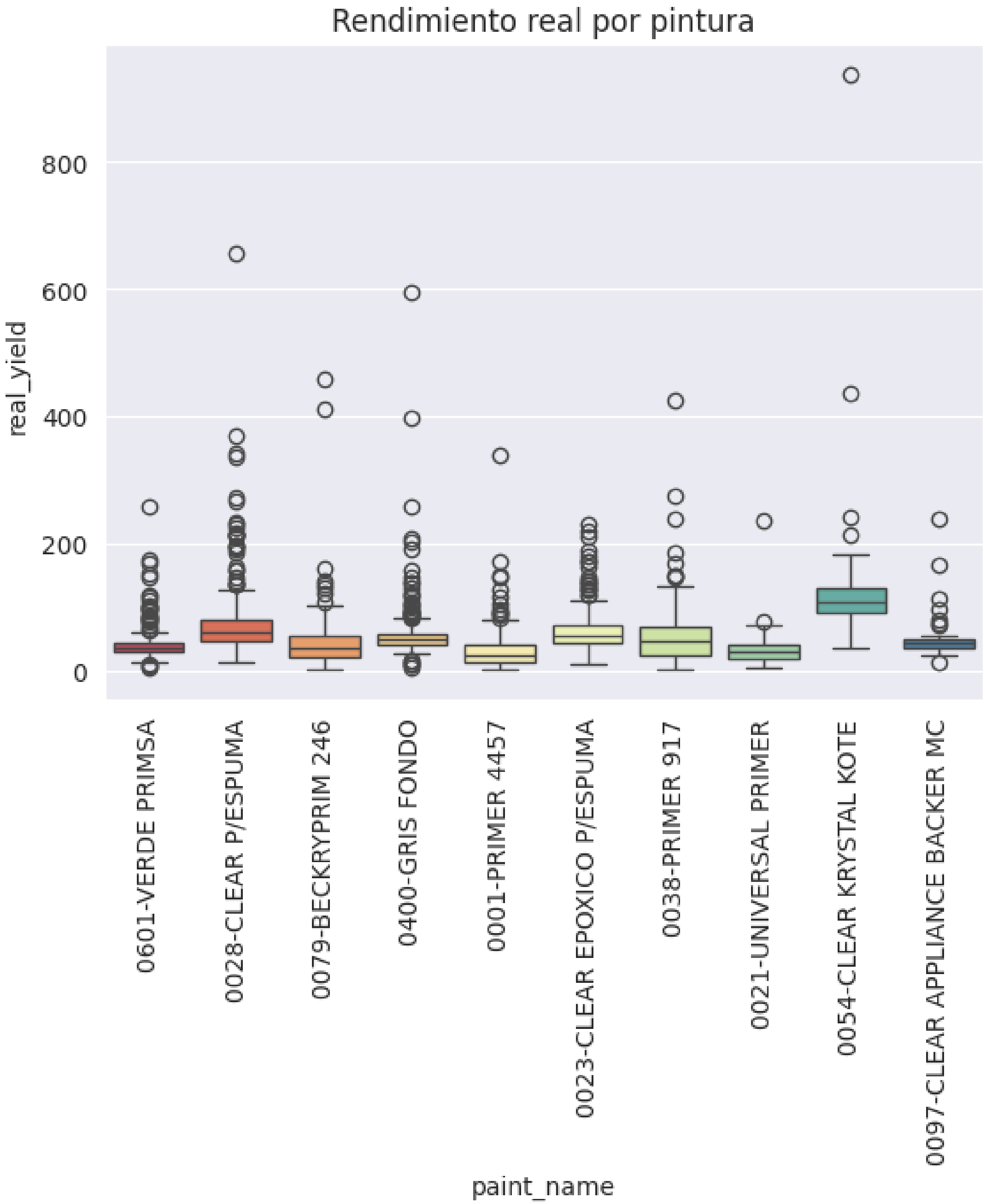
[69]:

```
                           expected_yield                               \
                           bottom_whisker        25%        50%
paint_name
0001-PRIMER 4457                58.267717  58.267717  58.267717
0021-UNIVERSAL PRIMER           54.330709  54.330709  54.330709
0023-CLEAR EPOXICO P/ESPUMA     44.291339  44.291339  44.291339
0028-CLEAR P/ESPUMA             44.440945  44.440945  44.440945
```

```
0038-PRIMER 917                   57.742782  57.742782  57.742782
0054-CLEAR KRYSTAL KOTE           18.673219  18.673219  18.673219
0079-BECKRYPRIM 246               67.637795  67.637795  67.637795
0097-CLEAR APPLIANCE BACKER MC    40.944882  40.944882  40.944882
0400-GRIS FONDO                   54.073034  54.073034  54.073034
0601-VERDE PRIMSA                 68.953881  68.953881  68.953881


                                     75%  top_whisker  IQR
paint_name
0001-PRIMER 4457                58.267717    58.267717  0.0
0021-UNIVERSAL PRIMER           54.330709    54.330709  0.0
0023-CLEAR EPOXICO P/ESPUMA     44.291339    44.291339  0.0
0028-CLEAR P/ESPUMA             44.440945    44.440945  0.0
0038-PRIMER 917                 57.742782    57.742782  0.0
0054-CLEAR KRYSTAL KOTE         18.673219    18.673219  0.0
0079-BECKRYPRIM 246             67.637795    67.637795  0.0
0097-CLEAR APPLIANCE BACKER MC  40.944882    40.944882  0.0
0400-GRIS FONDO                 54.073034    54.073034  0.0
0601-VERDE PRIMSA               68.953881    68.953881  0.0
```

[70]:

```
do_boxplot(most_used_paints_df, x='paint_name', y='real_yield', rotate=True,␣
 ↪title='Rendimiento real por pintura')
described_df.loc[:, ['real_yield']]
```



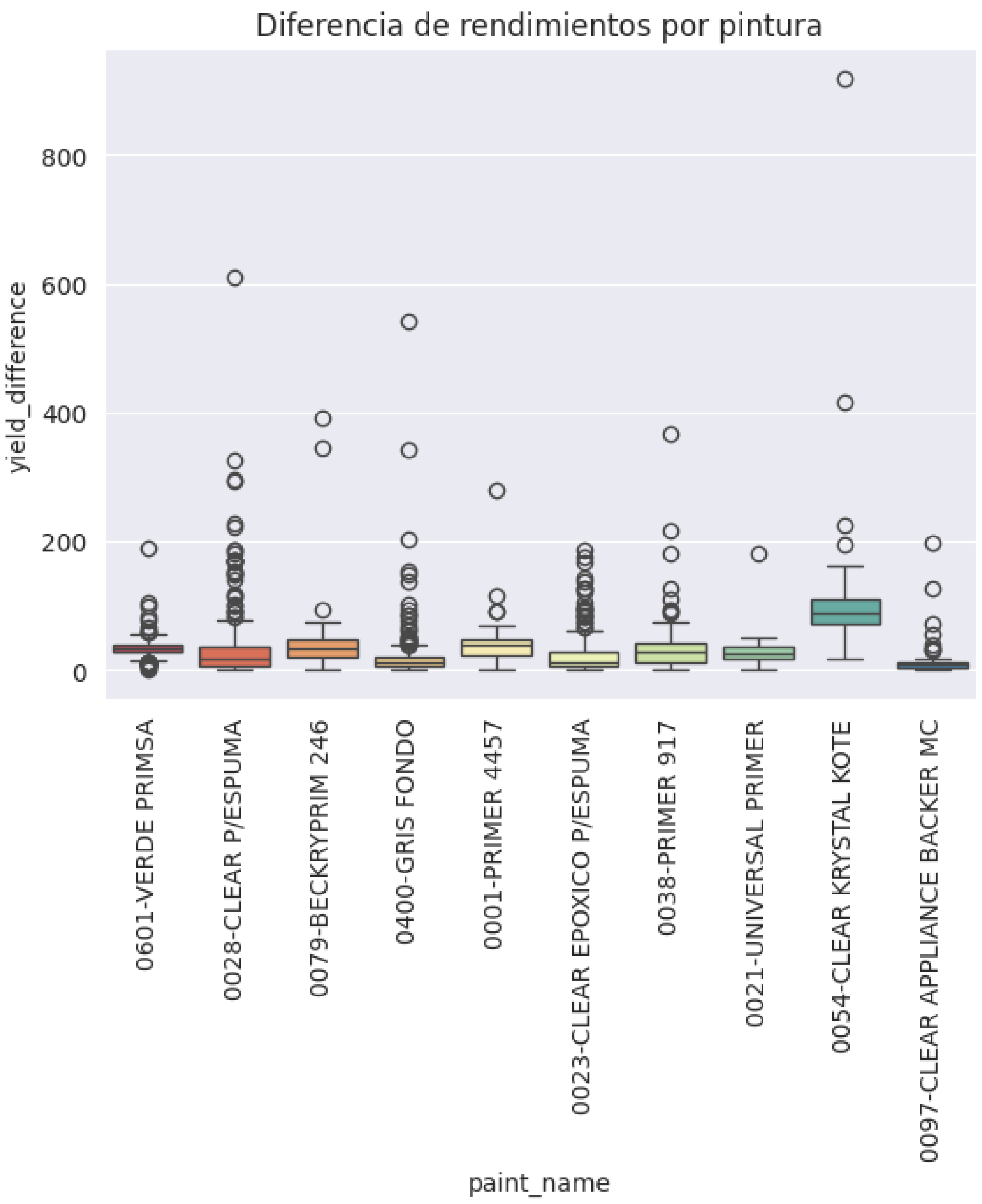
Rendimiento real por pintura

```
[70]:                                    real_yield                          \
                                     bottom_whisker        25%          50%
      paint_name
      0001-PRIMER 4457                     1.203259  12.890687    24.114641
      0021-UNIVERSAL PRIMER                5.154184  18.043164    28.783920
      0023-CLEAR EPOXICO P/ESPUMA         10.083014  42.559224    53.753736
      0028-CLEAR P/ESPUMA                 12.687590  45.201044    59.827029
```

```
0038-PRIMER 917                    1.341009  22.793488   44.943991
0054-CLEAR KRYSTAL KOTE           36.291259  91.004010  106.132498
0079-BECKRYPRIM 246                1.866800  21.901018   35.752701
0097-CLEAR APPLIANCE BACKER MC    15.795345  36.331300   44.484468
0400-GRIS FONDO                   17.890708  41.852807   47.856383
0601-VERDE PRIMSA                 11.558998  30.235953   35.187895


                                     75%  top_whisker        IQR
paint_name
0001-PRIMER 4457                 39.777104    80.106731  26.886418
0021-UNIVERSAL PRIMER            40.442618    74.041798  22.399454
0023-CLEAR EPOXICO P/ESPUMA      71.300011   114.411192  28.740787
0028-CLEAR P/ESPUMA              80.529614   133.522470  35.328570
0038-PRIMER 917                  67.856937   135.452111  45.063449
0054-CLEAR KRYSTAL KOTE         128.930680   185.820684  37.926669
0079-BECKRYPRIM 246              55.267491   105.317202  33.366474
0097-CLEAR APPLIANCE BACKER MC   50.021937    70.557892  13.690637
0400-GRIS FONDO                  57.827540    81.789639  15.974733
0601-VERDE PRIMSA                42.687257    61.364212  12.451303
```

[71]:

```
do_boxplot(most_used_paints_df, x='paint_name', y='yield_difference',␣
 ↪rotate=True,
           title='Diferencia de rendimientos por pintura')
described_df.loc[:, ['yield_difference']]
```



```
[71]:                           yield_difference                        \
                                  bottom_whisker        25%        50%
      paint_name
      0001-PRIMER 4457                  0.032277  23.133046  37.446399
      0021-UNIVERSAL PRIMER             0.793240  17.327479  25.814301
      0023-CLEAR EPOXICO P/ESPUMA       0.006838   5.538090  12.560462
      0028-CLEAR P/ESPUMA               0.116391   6.292136  17.477830
```

```
0038-PRIMER 917                         0.275460  12.394810  28.242272
0054-CLEAR KRYSTAL KOTE                17.618040  72.330792  87.459280
0079-BECKRYPRIM 246                     0.434319  20.050215  32.872026
0097-CLEAR APPLIANCE BACKER MC          0.673964   3.891186   7.600059
0400-GRIS FONDO                         0.089734   5.331463  10.018511
0601-VERDE PRIMSA                      11.576542  28.396231  34.178713


                                       75% top_whisker        IQR
paint_name
0001-PRIMER 4457                 46.405248   81.313551  23.272202
0021-UNIVERSAL PRIMER            36.592164   65.489191  19.264685
0023-CLEAR EPOXICO P/ESPUMA      28.851061   63.820517  23.312971
0028-CLEAR P/ESPUMA              36.088670   80.783470  29.796534
0038-PRIMER 917                  41.735019   85.745334  29.340210
0054-CLEAR KRYSTAL KOTE         110.257461  167.147465  37.926669
0079-BECKRYPRIM 246              46.572800   86.356678  26.522585
0097-CLEAR APPLIANCE BACKER MC   12.581466   25.616886   8.690280
0400-GRIS FONDO                  18.221631   37.556884  12.890168
0601-VERDE PRIMSA                39.609357   56.429045  11.213126
```

1.2.2 production_line

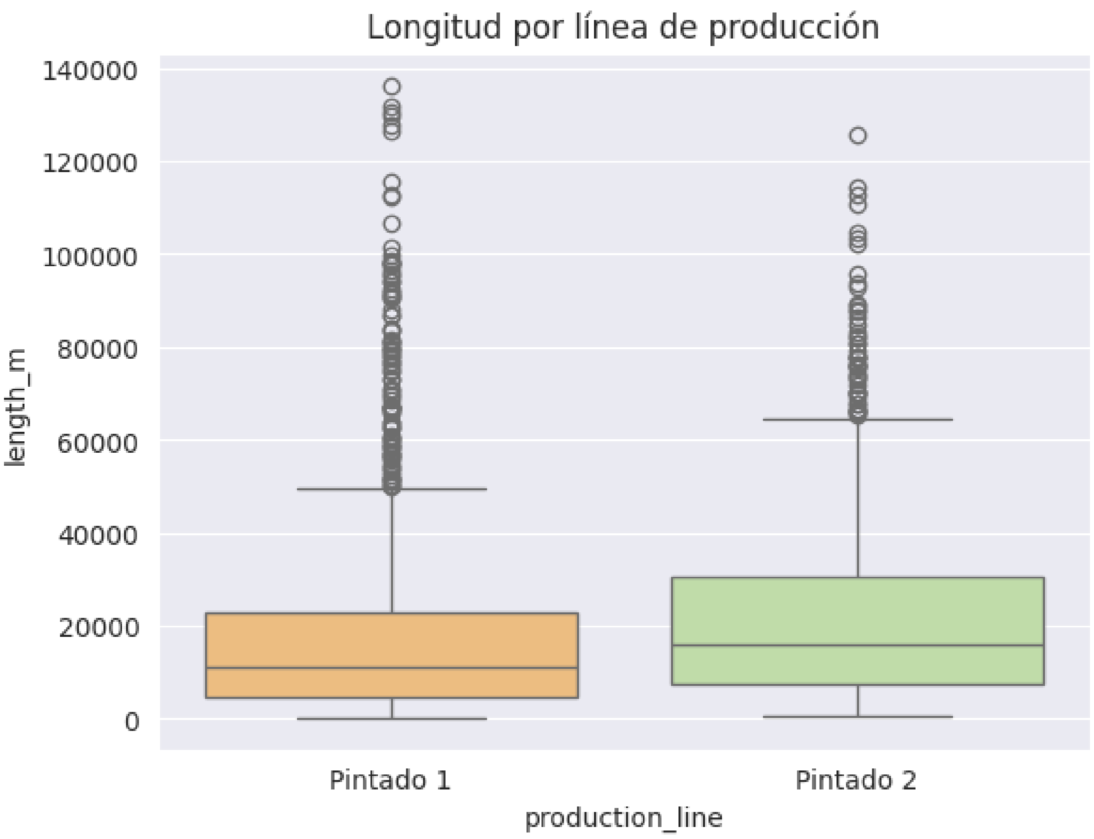
[72]:

```
per_production_line_described_df = \
most_used_paints_df.groupby('production_line').describe().stack(level=0,␣
 ↪future_stack=True)[
    ['75%', '50%', '25%', 'min', 'max']]
per_production_line_described_df = per_production_line_described_df.assign(
    IQR=lambda x: x['75%'] - x['25%'],
    top_whisker=lambda x: (x['75%'] + 1.5 * x['IQR']).combine(x['max'], min),
    bottom_whisker=lambda x: (x['25%'] - 1.5 * x['IQR']).combine(x['min'], max)
)[['bottom_whisker', '25%', '50%', '75%', 'top_whisker', 'IQR']].stack().
 ↪unstack(level=-2).unstack(level=-1)
```

```
/tmp/ipykernel_102683/48227659.py:2: FutureWarning: The default of
observed=False is deprecated and will be changed to True in a future version of
pandas. Pass observed=False to retain current behavior or observed=True to adopt
the future default and silence this warning.
  most_used_paints_df.groupby('production_line').describe().stack(level=0,
future_stack=True)[
```

[73]:

```
do_boxplot(most_used_paints_df, x='production_line', y='length_m',␣
 ↪title='Longitud por línea de producción')
per_production_line_described_df.loc[:, ['length_m']]
```

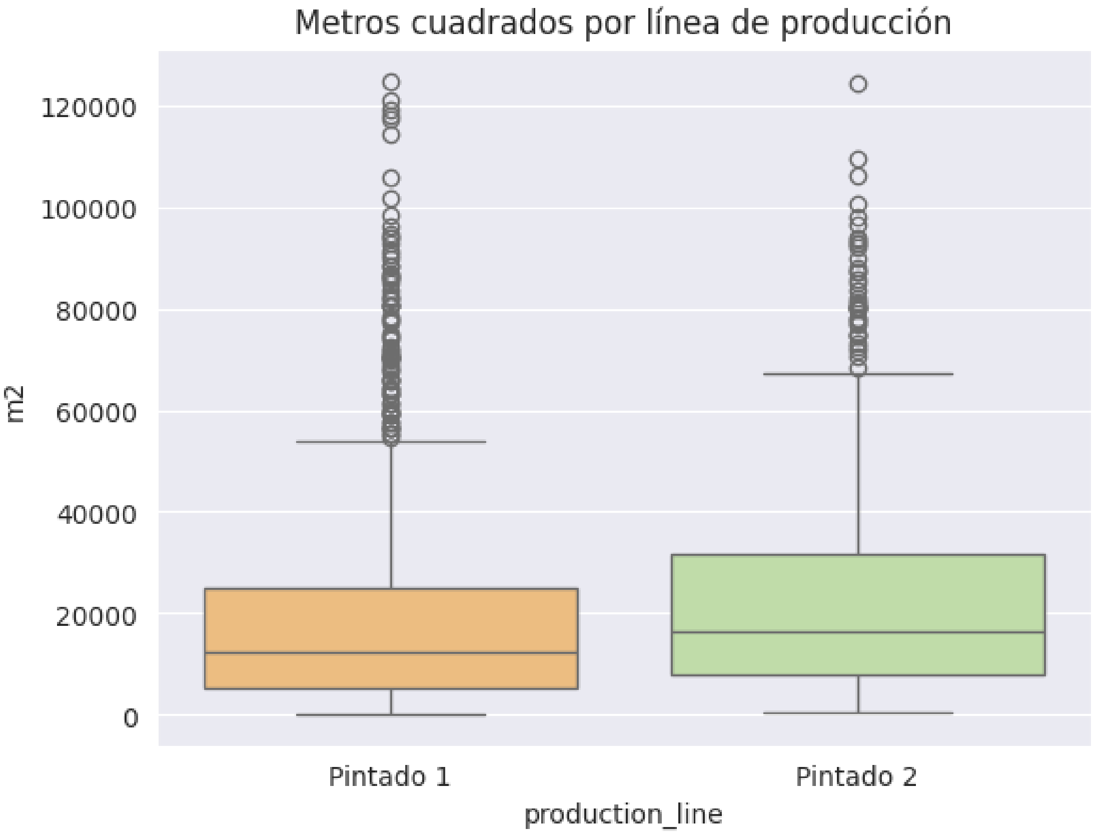


[73]:

```
                      length_m                                            \
                bottom_whisker      25%      50%       75% top_whisker
production_line
Pintado 1                234.0  4758.75  10933.5  22898.75    50108.75
Pintado 2                367.0  7485.00  15938.0  30531.00    65100.00


                     IQR
production_line
Pintado 1        18140.0
Pintado 2        23046.0
```

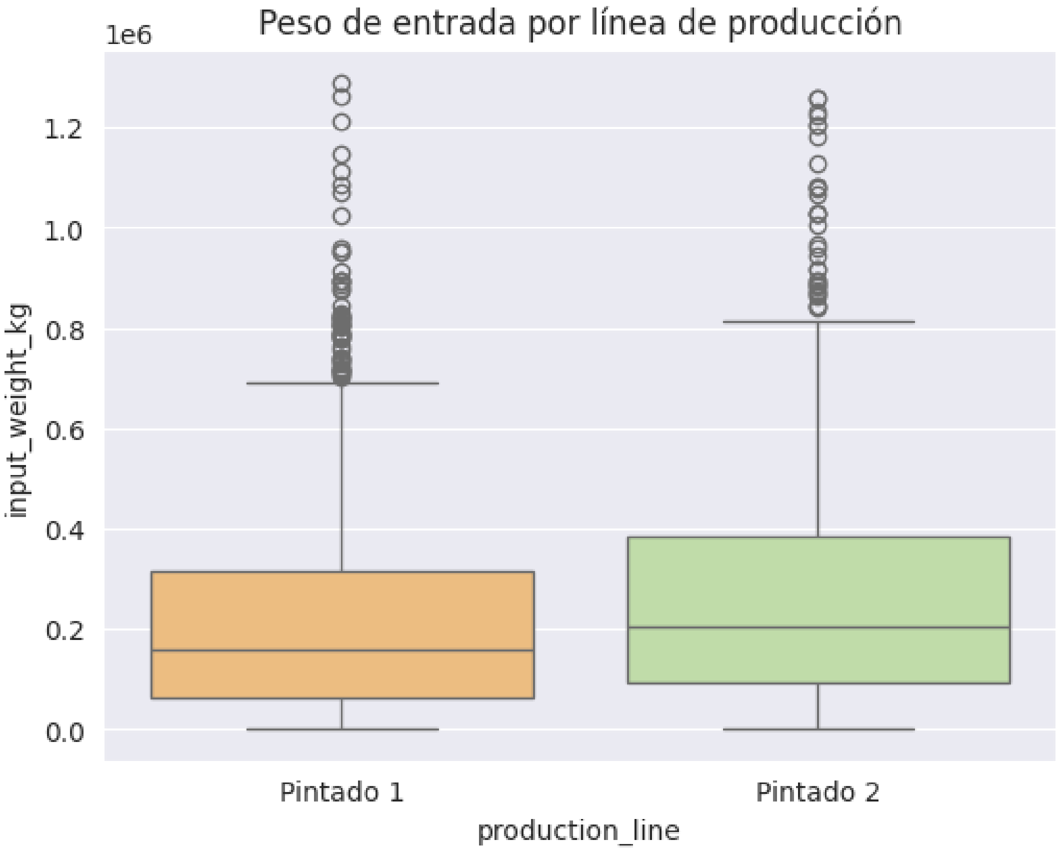
[74]:

```
do_boxplot(most_used_paints_df, x='production_line', y='m2', title='Metros␣
 ↪cuadrados por línea de producción')
per_production_line_described_df.loc[:, ['m2']]
```



[74]:
```
                                 m2                                                 \
                     bottom_whisker        25%        50%         75%  top_whisker
production_line
Pintado 1                   214.812  5489.5995  12431.615  24989.8900  54240.32575
Pintado 2                   410.640  7763.4650  16462.610  31819.4055  67903.31625


                        IQR
production_line
Pintado 1        19500.2905
Pintado 2        24055.9405
```

[75]:
```
do_boxplot(most_used_paints_df, x='production_line', y='input_weight_kg',
           title='Peso de entrada por línea de producción')
per_production_line_described_df.loc[:, ['input_weight_kg']]
```

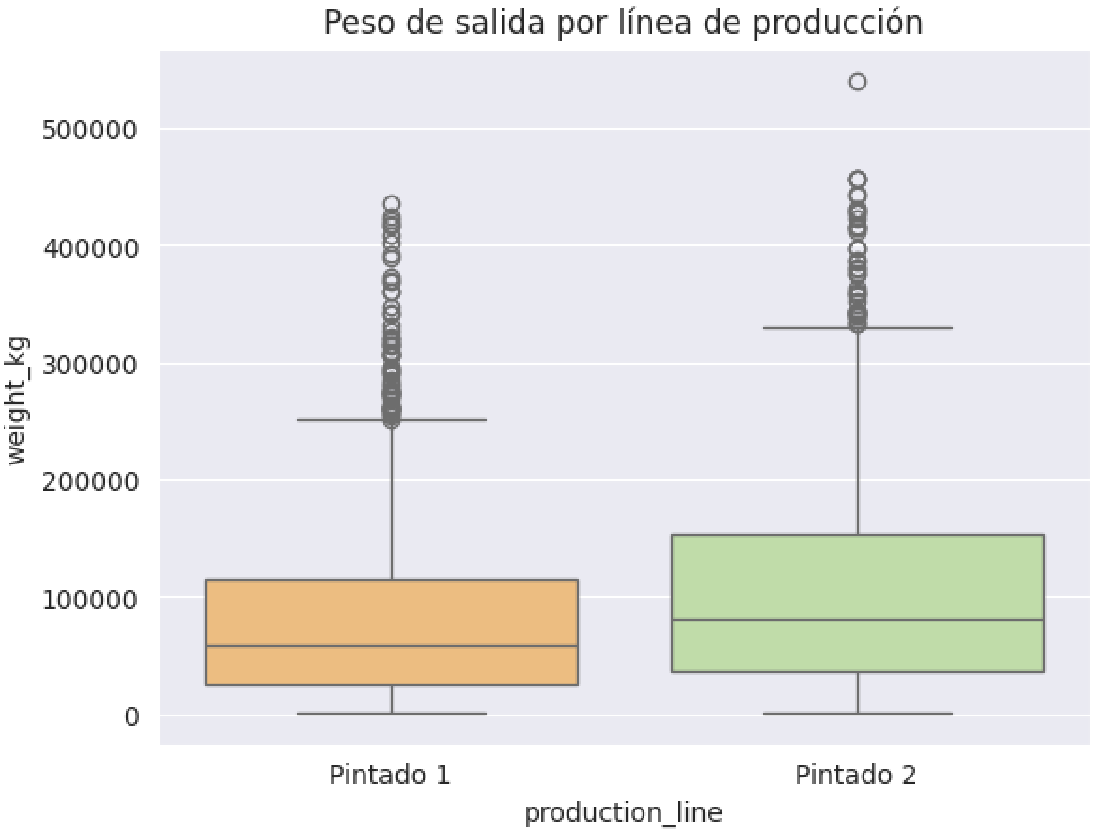


[75]:

```
                 input_weight_kg                                           \
                  bottom_whisker       25%       50%         75% top_whisker
production_line
Pintado 1                 2251.0  62201.25  159995.0   317124.25   699508.75
Pintado 2                 2010.0  93141.50  202737.0   382832.00   817367.75


                       IQR
production_line
Pintado 1         254923.0
Pintado 2         289690.5
```

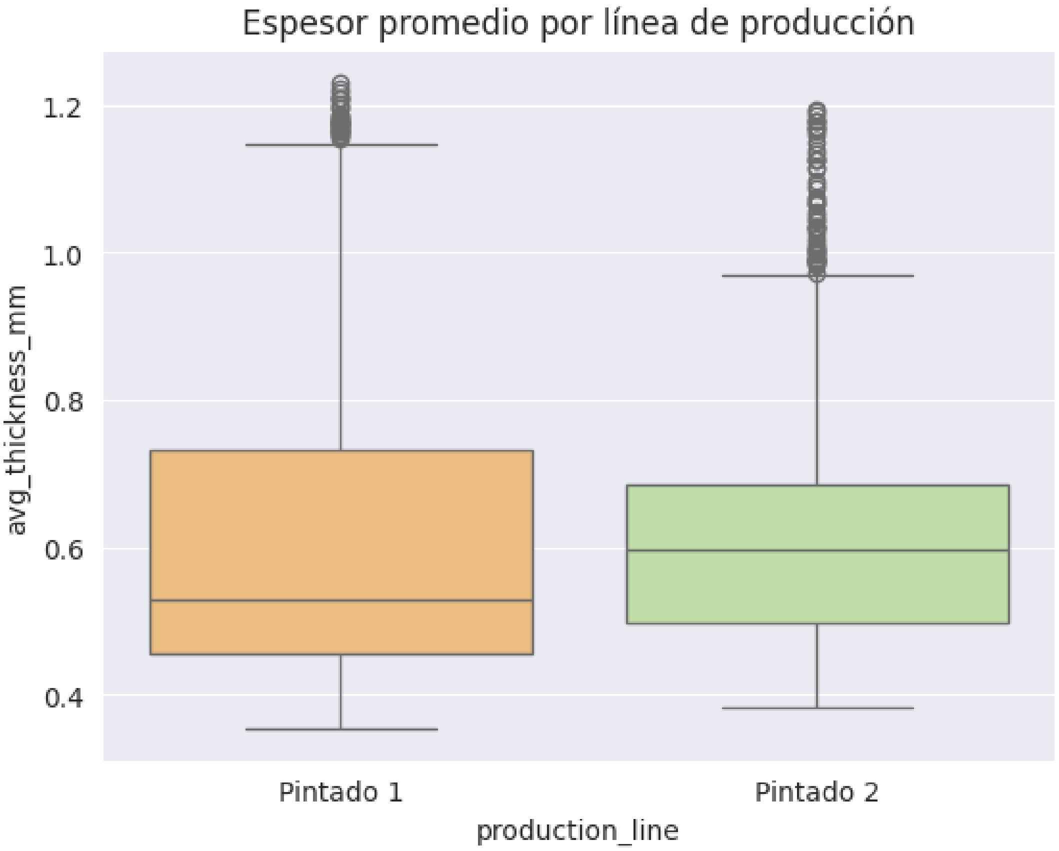
[76]:

```
do_boxplot(most_used_paints_df, x='production_line', y='weight_kg', title='Peso␣
 ↪de salida por línea de producción')
per_production_line_described_df.loc[:, ['weight_kg']]
```



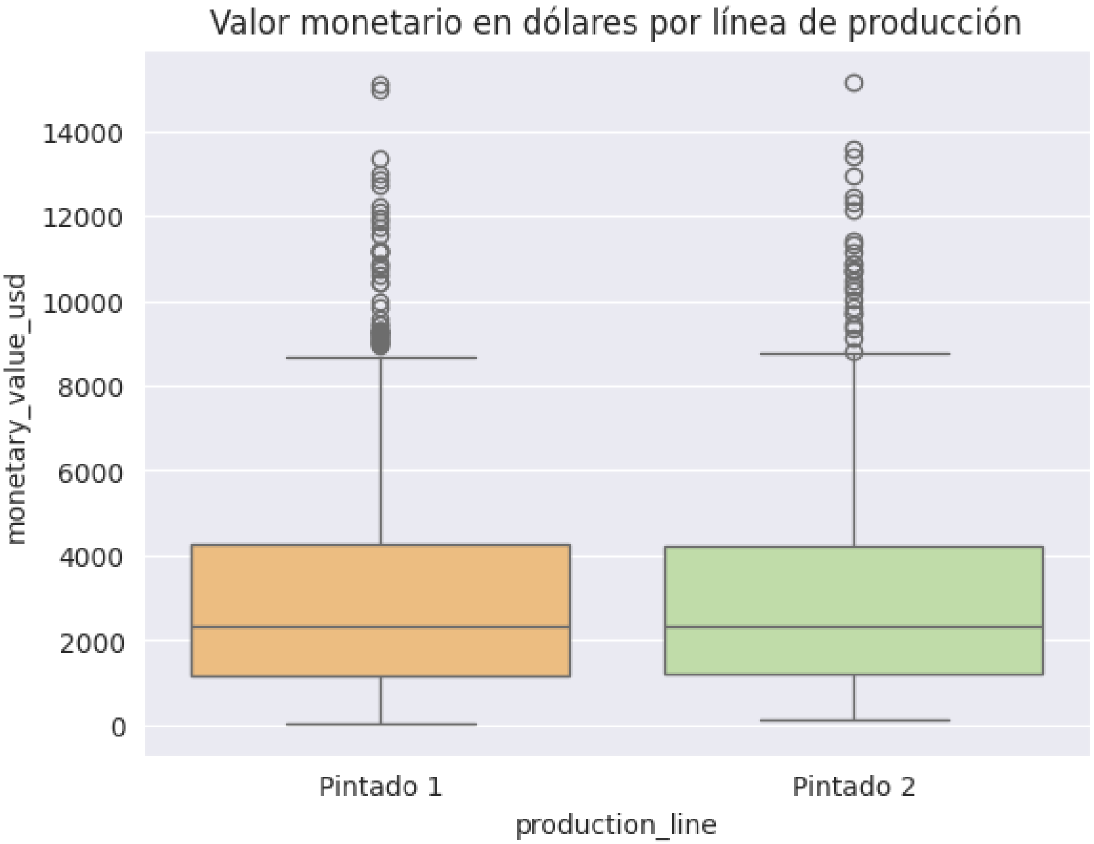
[76]:
```
                     weight_kg                                               \
                bottom_whisker      25%      50%        75% top_whisker   
production_line                                                           
Pintado 1               1790.0  24589.0  58492.5  115039.75  250715.875   
Pintado 2               1630.0  35993.0  81251.0  153463.50  329669.250   


                       IQR  
production_line             
Pintado 1         90450.75  
Pintado 2        117470.50  
```

[77]:
```
do_boxplot(most_used_paints_df, x='production_line', y='avg_thickness_mm',
           title='Espesor promedio por línea de producción')
per_production_line_described_df.loc[:, ['avg_thickness_mm']]
```



[77]:
```
                avg_thickness_mm                                          \
                  bottom_whisker       25%       50%       75% top_whisker
production_line
Pintado 1               0.354000  0.454679  0.528294  0.732490    1.149208
Pintado 2               0.380936  0.496677  0.596337  0.685927    0.969803


                      IQR
production_line
Pintado 1        0.277812
Pintado 2        0.189251
```

[78]:
```
do_boxplot(most_used_paints_df, x='production_line', y='monetary_value_usd',
           title='Valor monetario en dólares por línea de producción')
per_production_line_described_df.loc[:, ['monetary_value_usd']]
```

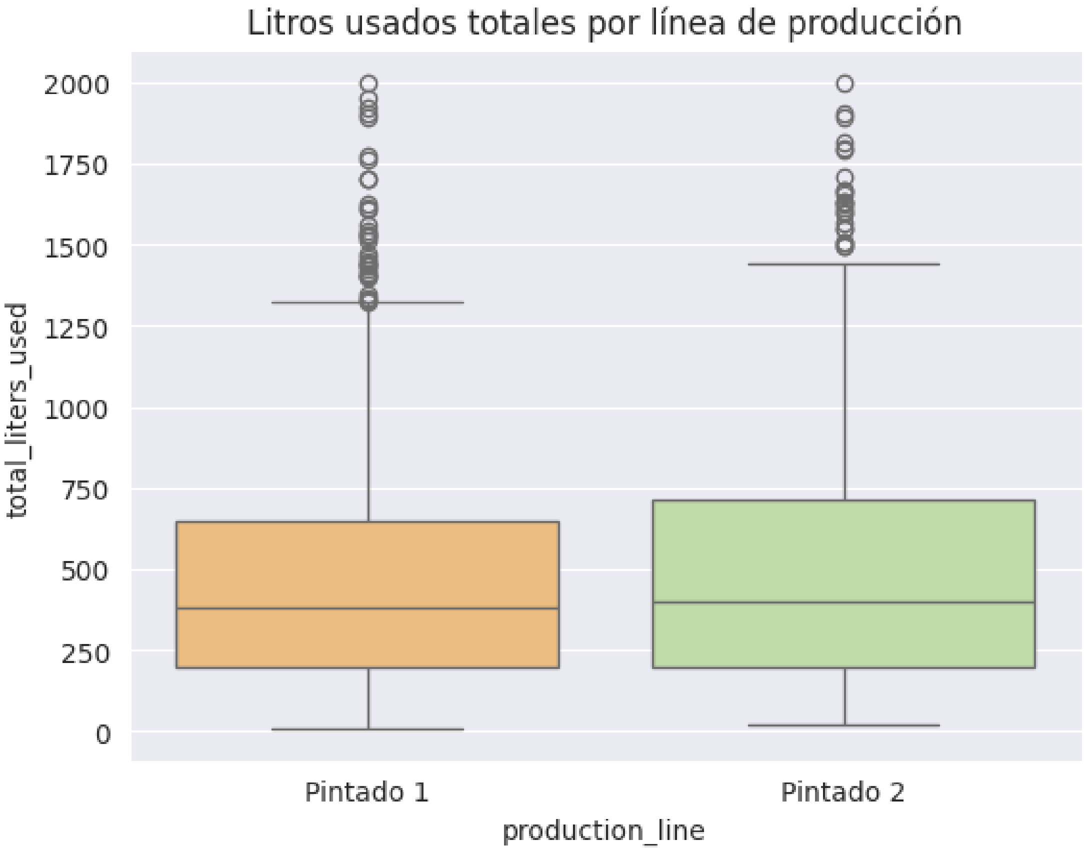


[78]:

```
                   monetary_value_usd                                            \
                       bottom_whisker      25%     50%        75% top_whisker
production_line
Pintado 1                       48.00  1144.96  2312.0  4256.0000  8922.56000
Pintado 2                      112.29  1189.67  2334.1  4236.7275  8807.31375


                         IQR
production_line
Pintado 1          3111.0400
Pintado 2          3047.0575
```

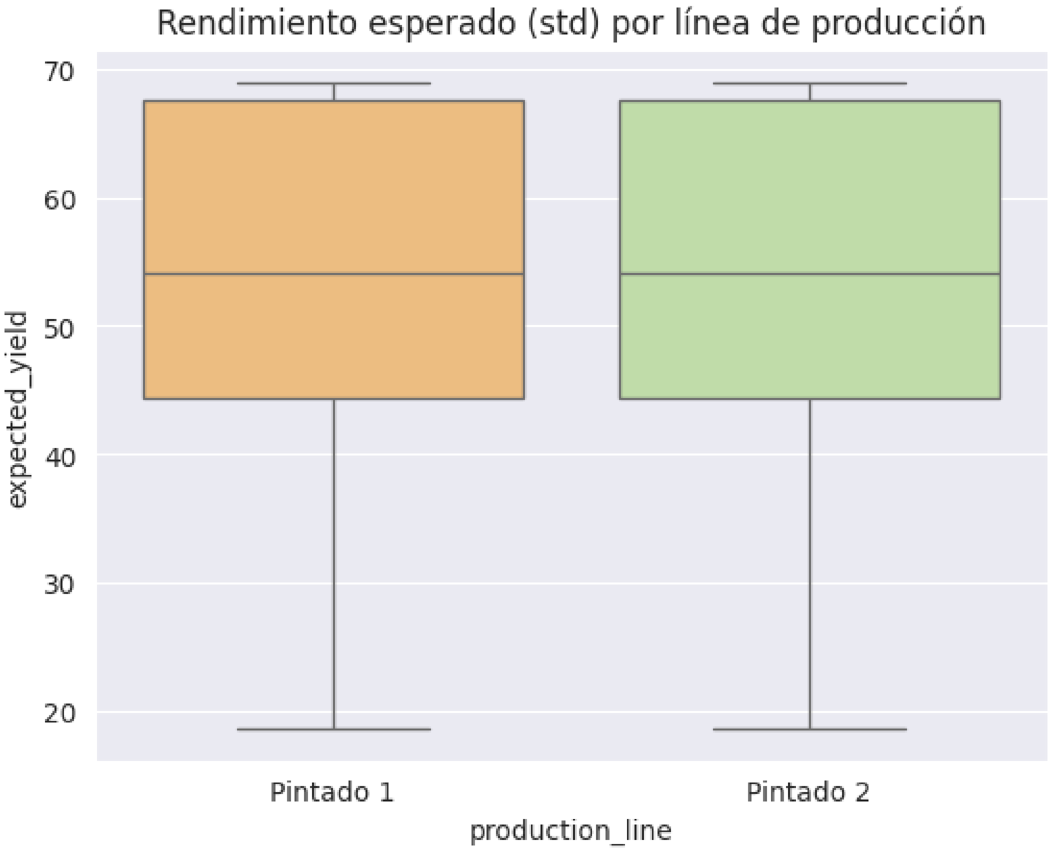
[79]:

```
do_boxplot(most_used_paints_df, x='production_line', y='total_liters_used',
           title='Litros usados totales por línea de producción')
per_production_line_described_df.loc[:, ['total_liters_used']]
```



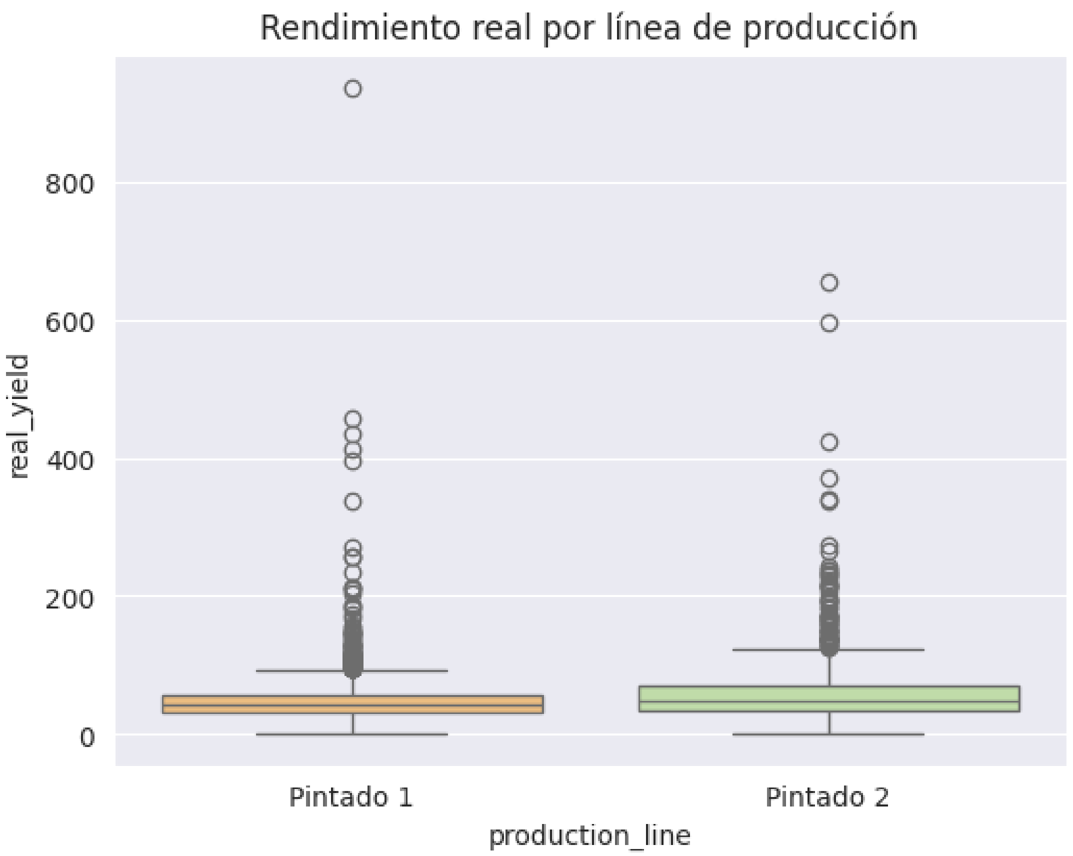
[79]:
```
                      total_liters_used
                         bottom_whisker    25%    50%      75% top_whisker      IQR
production_line
Pintado 1                          10.0  200.0  380.0  650.000   1325.0000  450.000
Pintado 2                          19.0  200.0  400.0  711.375   1478.4375  511.375
```

[80]:
```
do_boxplot(most_used_paints_df, x='production_line', y='expected_yield',
           title='Rendimiento esperado (std) por línea de producción')
per_production_line_described_df.loc[:, ['expected_yield']]
```



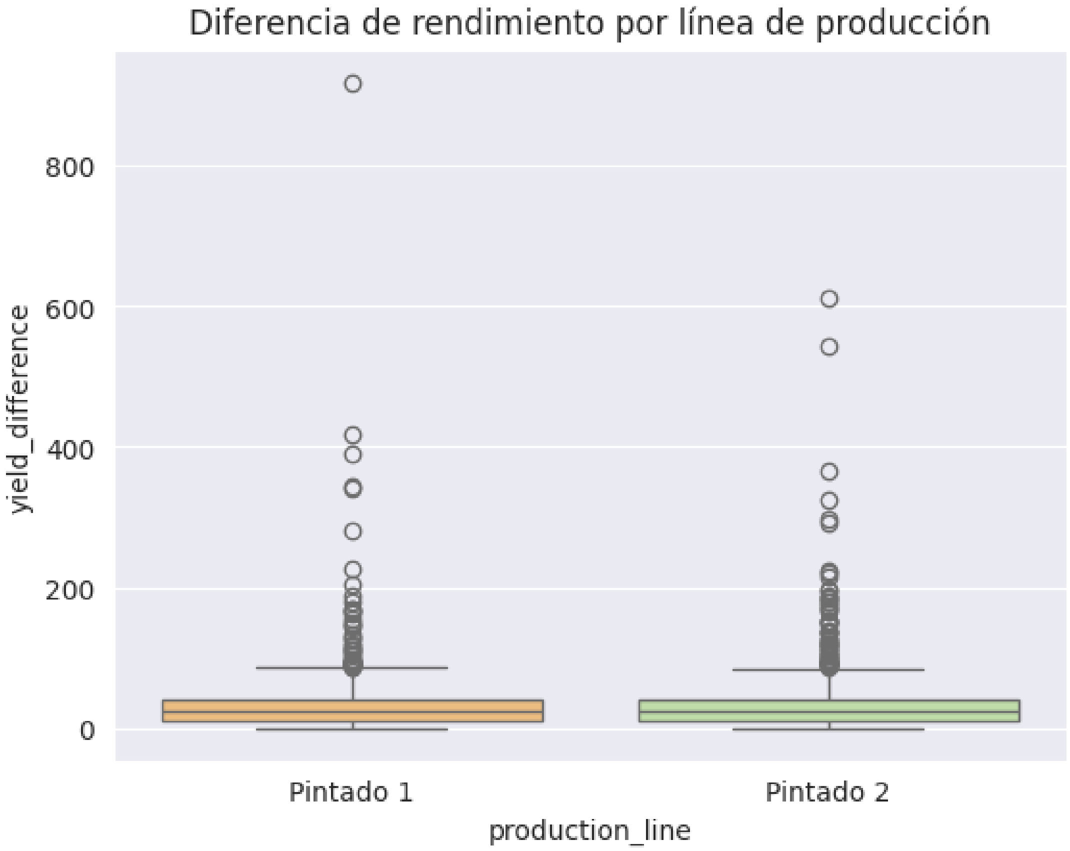
```
[80]:                  expected_yield                                            \
                       bottom_whisker        25%        50%        75% top_whisker
      production_line
      Pintado 1             18.673219  44.440945  54.073034  67.637795   68.953881
      Pintado 2             18.673219  44.440945  54.073034  67.637795   68.953881


                             IQR
      production_line
      Pintado 1         23.19685
      Pintado 2         23.19685
```

```
[81]: do_boxplot(most_used_paints_df, x='production_line', y='real_yield',␣
        ↪title='Rendimiento real por línea de producción')
      per_production_line_described_df.loc[:, ['real_yield']]
```



[81]:
```
                        real_yield                                              \
                    bottom_whisker        25%        50%        75% top_whisker
production_line
Pintado 1                 1.203259  29.902978  41.594132  55.428842   93.717637
Pintado 2                 1.866800  34.965494  48.344787  71.076351  125.242637


                       IQR
production_line
Pintado 1        25.525863
Pintado 2        36.110858
```

[82]:
```
do_boxplot(most_used_paints_df, x='production_line', y='yield_difference',
           title='Diferencia de rendimiento por línea de producción')
per_production_line_described_df.loc[:, ['yield_difference']]
```



[82]:

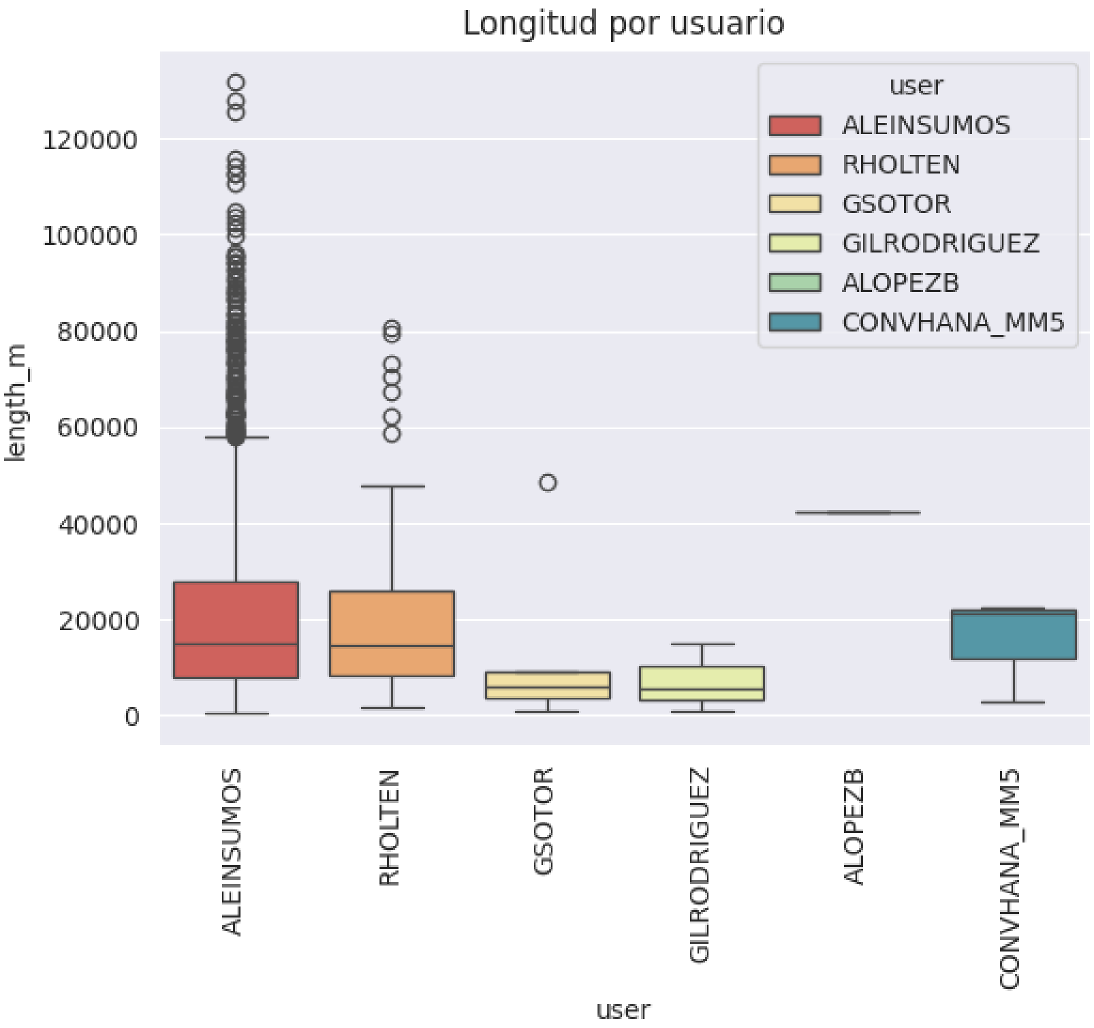
```
                     yield_difference                                                    \
                       bottom_whisker        25%        50%        75% top_whisker
production_line
Pintado 1                    0.089734   9.612517  24.940929  40.372667   86.512894
Pintado 2                    0.006838  10.167785  24.661900  40.366754   85.665209


                           IQR
production_line
Pintado 1            30.760151
Pintado 2            30.198970
```

1.2.3 user

[83]:

```
per_user_described_df = most_used_paints_df.groupby('user').describe().
  ↪stack(level=0, future_stack=True)[
    ['75%', '50%', '25%', 'min', 'max']]
per_user_described_df = per_user_described_df.assign(
    IQR=lambda x: x['75%'] - x['25%'],
    top_whisker=lambda x: (x['75%'] + 1.5 * x['IQR']).combine(x['max'], min),
    bottom_whisker=lambda x: (x['25%'] - 1.5 * x['IQR']).combine(x['min'], max)
)[['bottom_whisker', '25%', '50%', '75%', 'top_whisker', 'IQR']].stack().
  ↪unstack(level=-2).unstack(level=-1)
```

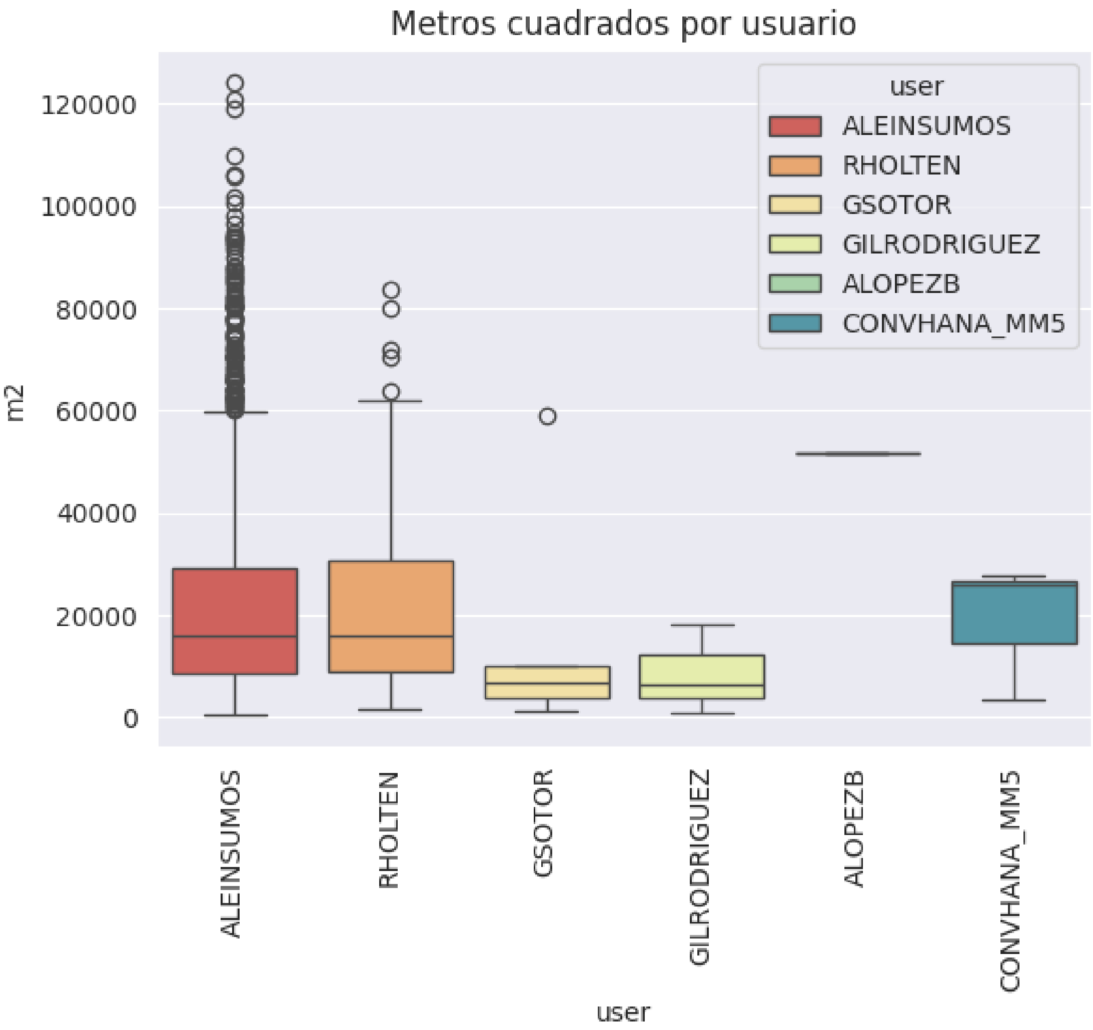
[84]:
```
do_boxplot(most_used_paints_df, x='user', y='length_m', title='Longitud por␣
  ↪usuario', rotate=True)
per_user_described_df.loc[:, ['length_m']]
```



[84]:
```
                       length_m
                 bottom_whisker      25%      50%      75% top_whisker      IQR
user
ALEINSUMOS                426.0   7652.0  15006.0  27828.0    58092.00  20176.0
ALOPEZB                 42180.0  42180.0  42180.0  42180.0    42180.00      0.0
CONVHANA_MM5             2687.0  11861.5  21036.0  21752.0    22468.00   9890.5
```

```
GILRODRIGUEZ          718.0    3027.5    5337.0  10129.5     14922.00   7102.0
GSOTOR                919.0    3365.0    6006.0   8974.0     17387.50   5609.0
RHOLTEN              1394.0    8137.0   14407.5  25744.5     52155.75  17607.5
```

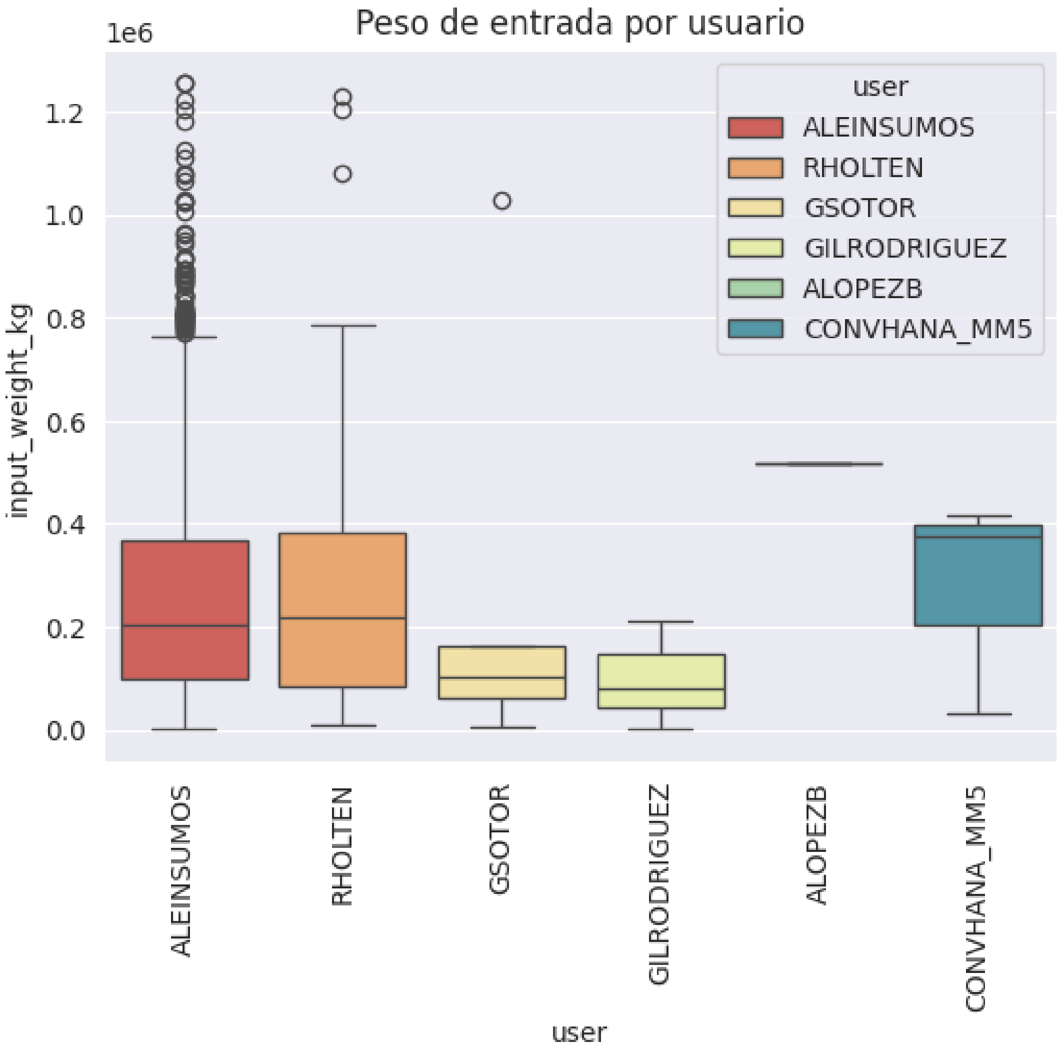
[85]:
```
do_boxplot(most_used_paints_df, x='user', y='m2', title='Metros cuadrados por␣
↪usuario', rotate=True)
per_user_described_df.loc[:, ['m2']]
```



[85]:
```
                          m2                                              \
              bottom_whisker           25%           50%           75%
user
ALEINSUMOS         522.27600   8431.172969   16100.89500   29056.5870
ALOPEZB          51542.65600  51542.656000   51542.65600   51542.6560
CONVHANA_MM5      3288.88800  14599.050000   25909.21200   26783.4480
```

```
GILRODRIGUEZ      877.39600   3697.715500   6518.03500  12377.6845
GSOTOR           1134.34927   3655.726000   6791.54961  10220.7600
RHOLTEN          1653.28400   8872.945250  16005.20800  30736.4850


               top_whisker           IQR
user
ALEINSUMOS    59994.708047  20625.414031
ALOPEZB       51542.656000      0.000000
CONVHANA_MM5  27657.684000  12184.398000
GILRODRIGUEZ  18237.334000   8679.969000
GSOTOR        20068.311000   6565.034000
RHOLTEN       63531.794625  21863.539750
```

[86]:
```
do_boxplot(most_used_paints_df, x='user', y='input_weight_kg', title='Peso de␣
 ↪entrada por usuario', rotate=True)
per_user_described_df.loc[:, ['input_weight_kg']]
```



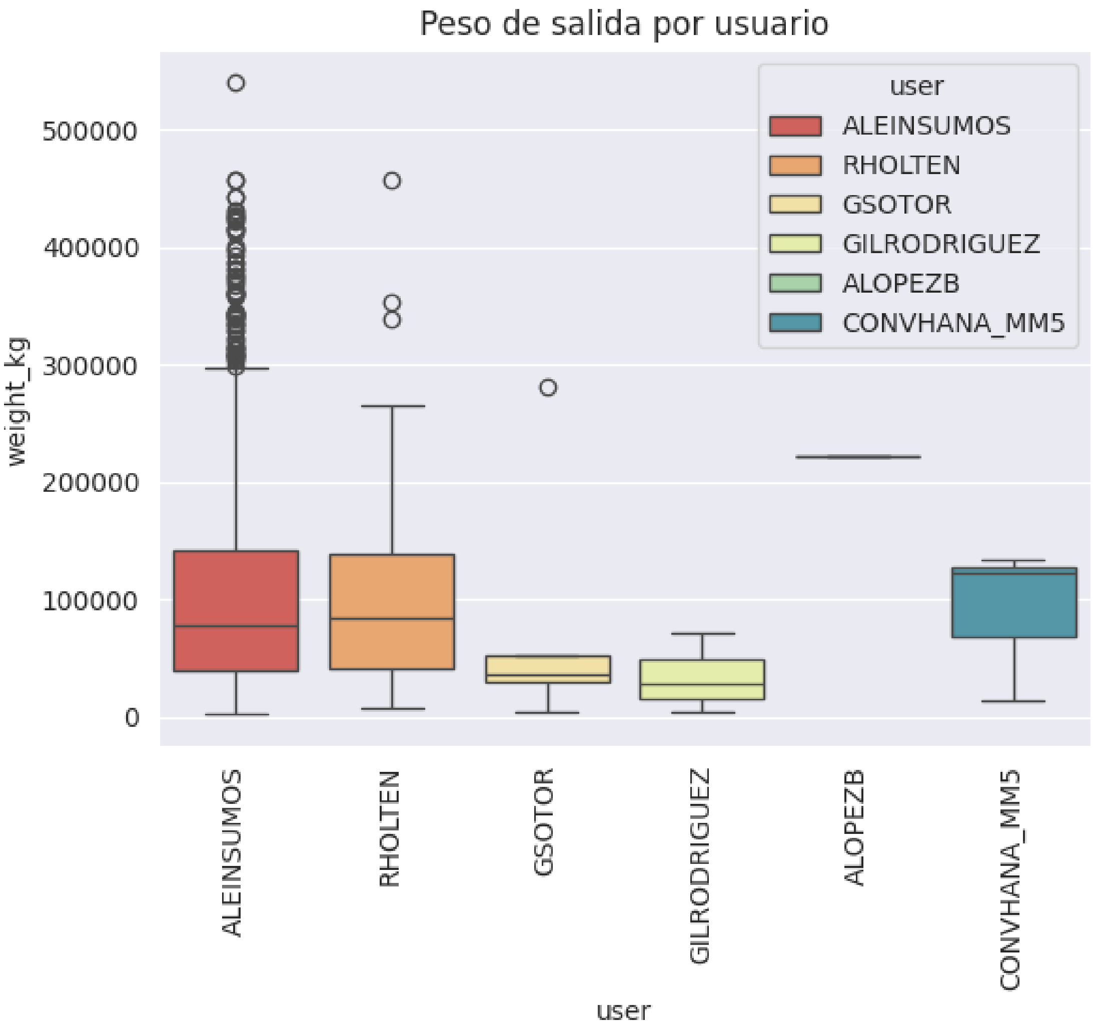
[86]:

```
             input_weight_kg                                                  \
              bottom_whisker        25%            50%       75% top_whisker
user
ALEINSUMOS            2700.0   99840.00  202872.072688  367440.0  768840.000
ALOPEZB             517344.0  517344.00  517344.000000  517344.0  517344.000
CONVHANA_MM5         31930.0  203901.50  375873.000000  396410.5  416948.000
GILRODRIGUEZ          3515.0   42677.50   81840.000000  146465.0  211090.000
GSOTOR                5037.0   60980.00  103104.000000  163198.0  316525.000
RHOLTEN               8239.0   85741.25  220118.500000  381794.5  825874.375


                    IQR
user
ALEINSUMOS    267600.00
```

```
ALOPEZB              0.00
CONVHANA_MM5    192509.00
GILRODRIGUEZ    103787.50
GSOTOR          102218.00
RHOLTEN         296053.25
```

[87]:

```
do_boxplot(most_used_paints_df, x='user', y='weight_kg', title='Peso de salida␣
 ↪por usuario', rotate=True)
per_user_described_df.loc[:, ['weight_kg']]
```



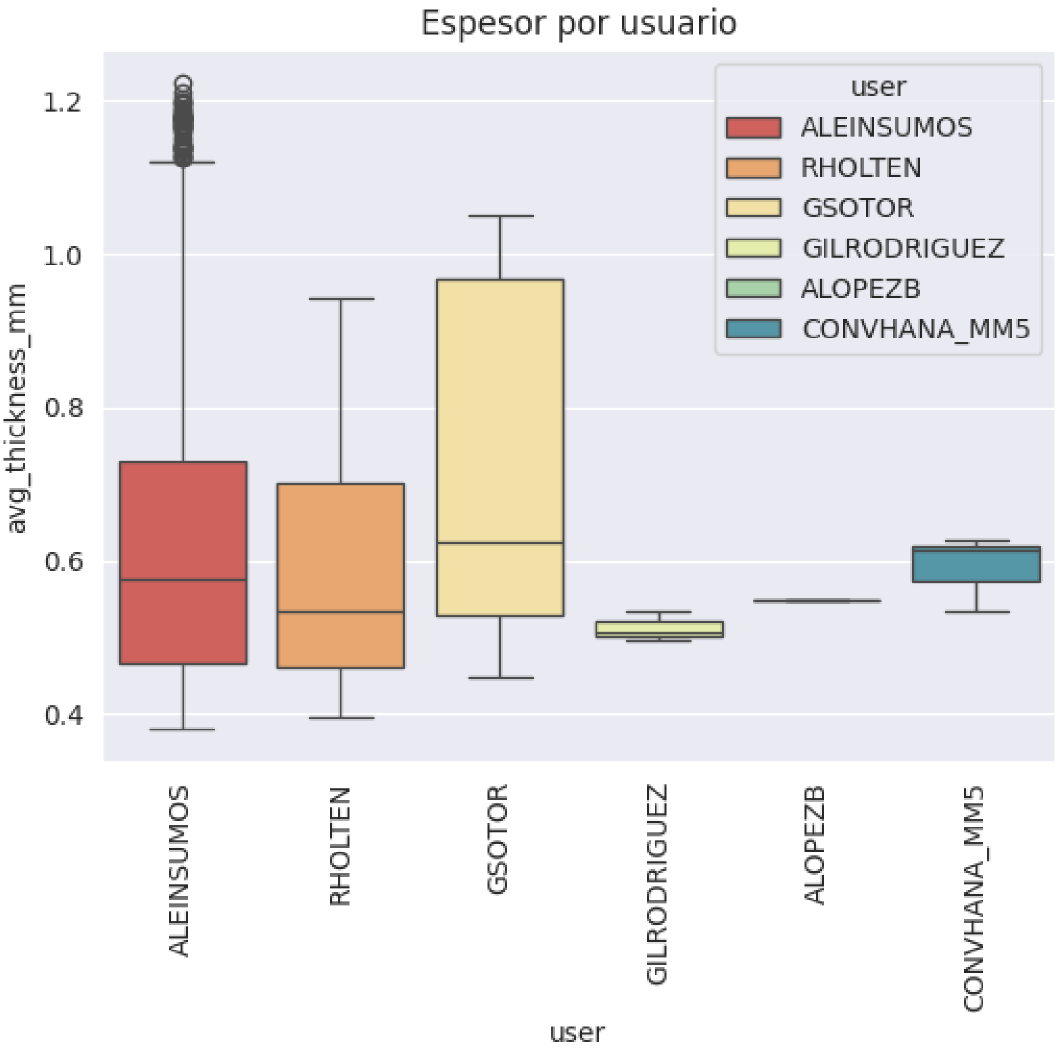
[87]:

```
                   weight_kg                                              \
              bottom_whisker       25%      50%       75% top_whisker
user
ALEINSUMOS            2672.0  38631.00  78238.0  142435.0  298141.000
```

```
ALOPEZB         222250.0  222250.00  222250.0  222250.0  222250.000
CONVHANA_MM5     13975.0   68346.50  122718.0  127868.0  133018.000
GILRODRIGUEZ      3465.0   15512.50   27560.0   49285.0   71010.000
GSOTOR            4747.0   30355.00   36268.0   51894.0   84202.500
RHOLTEN           6994.0   40648.75   84375.0  138321.0  284829.375


                    IQR
user
ALEINSUMOS    103804.00
ALOPEZB            0.00
CONVHANA_MM5   59521.50
GILRODRIGUEZ   33772.50
GSOTOR         21539.00
RHOLTEN        97672.25
```

[88]:

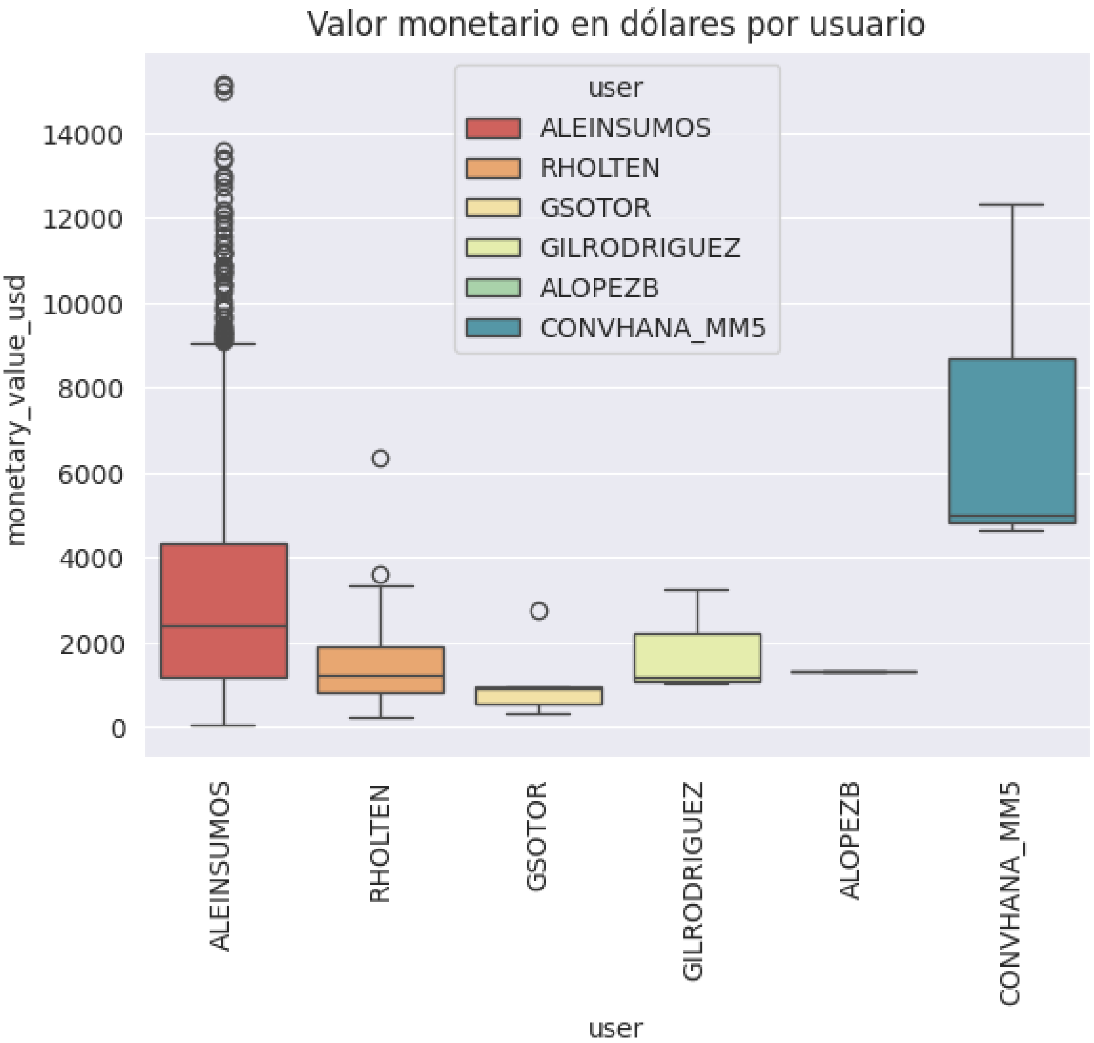
```
do_boxplot(most_used_paints_df, x='user', y='avg_thickness_mm', title='Espesor␣
  ↪por usuario', rotate=True)
per_user_described_df.loc[:, ['avg_thickness_mm']]
```



```
[88]:              avg_thickness_mm                                          \
                     bottom_whisker       25%       50%       75% top_whisker
     user
     ALEINSUMOS            0.380936  0.465500  0.575320  0.728250    1.122375
     ALOPEZB               0.549333  0.549333  0.549333  0.549333    0.549333
     CONVHANA_MM5          0.533000  0.573000  0.613000  0.619775    0.626550
     GILRODRIGUEZ          0.495267  0.500233  0.505200  0.519850    0.534500
     GSOTOR                0.447333  0.529000  0.623858  0.968090    1.050400
     RHOLTEN               0.397000  0.459926  0.533752  0.702350    0.942233


                       IQR
     user
     ALEINSUMOS    0.262750
```

```
ALOPEZB        0.000000
CONVHANA_MM5   0.046775
GILRODRIGUEZ   0.019617
GSOTOR         0.439090
RHOLTEN        0.242424
```

[89]:
```python
do_boxplot(most_used_paints_df, x='user', y='monetary_value_usd', title='Valor␣
↪monetario en dólares por usuario',
           rotate=True)
per_user_described_df.loc[:, ['monetary_value_usd']]
```

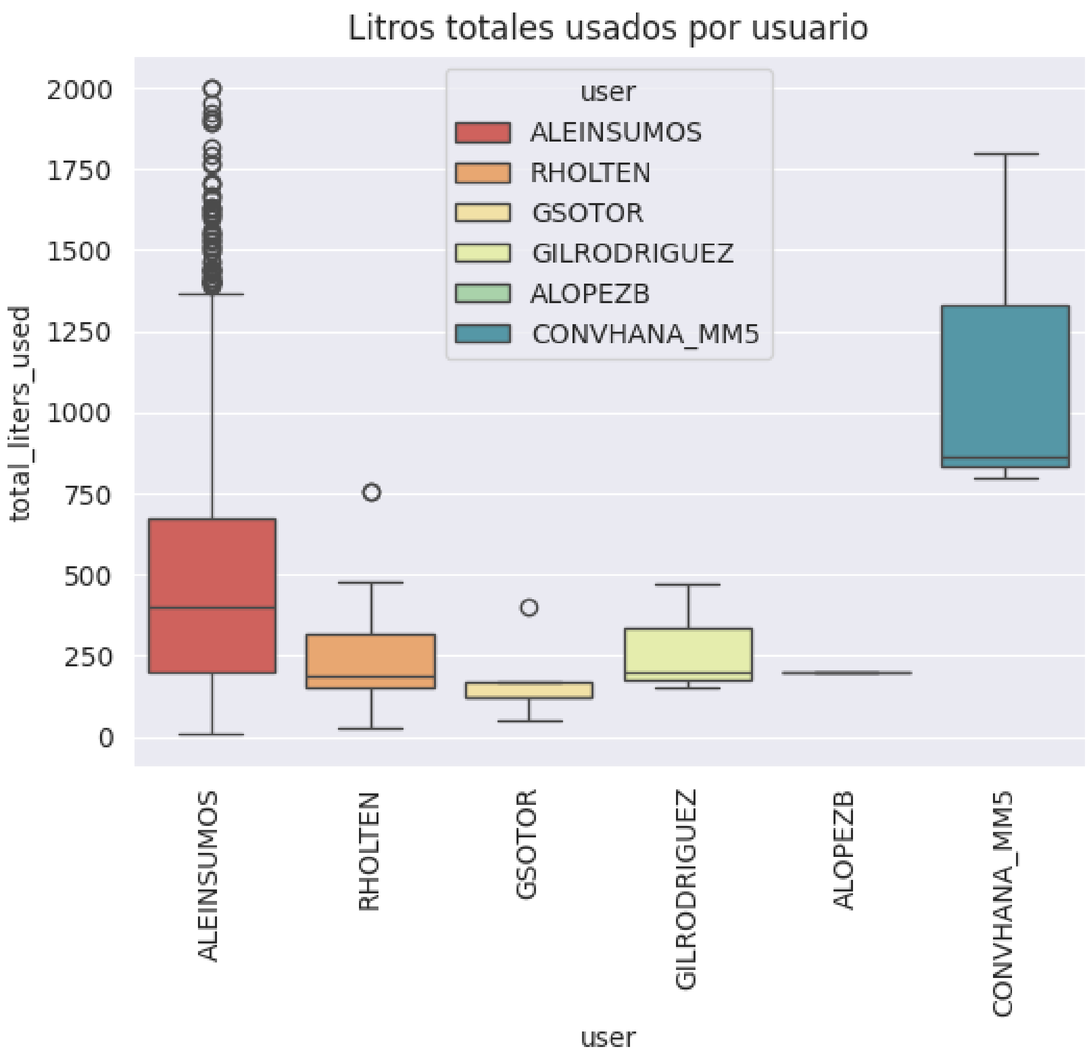


[89]:
```
                monetary_value_usd                                        \
                    bottom_whisker       25%        50%       75% top_whisker
user
```

```
ALEINSUMOS                48.0  1188.00  2369.220  4336.20    9058.500
ALOPEZB                 1288.0  1288.00  1288.000  1288.00    1288.000
CONVHANA_MM5            4624.0  4811.85  4999.700  8664.85   12330.000
GILRODRIGUEZ            1038.0  1094.00  1150.000  2201.20    3252.400
GSOTOR                   295.5   525.60   914.600   930.00    1536.600
RHOLTEN                  220.5   812.20  1228.105  1907.55    3550.575


                  IQR
user
ALEINSUMOS    3148.20
ALOPEZB          0.00
CONVHANA_MM5  3853.00
GILRODRIGUEZ  1107.20
GSOTOR         404.40
RHOLTEN       1095.35
```

[90]:

```
do_boxplot(most_used_paints_df, x='user', y='total_liters_used', title='Litros␣
 ↪totales usados por usuario', rotate=True)
per_user_described_df.loc[:, ['total_liters_used']]
```

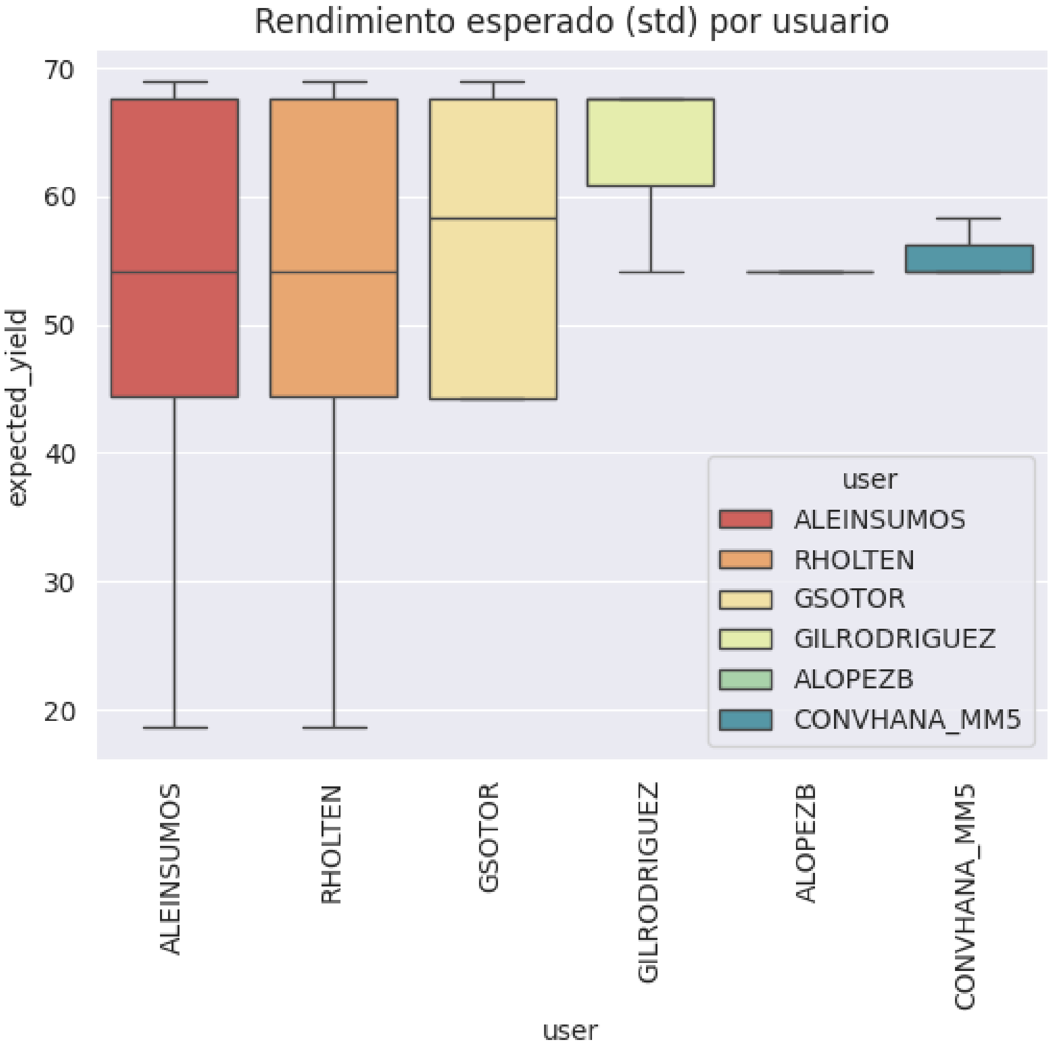


```
[90]:            total_liters_used                                               \
                    bottom_whisker       25%     50%      75% top_whisker
      user
      ALEINSUMOS              10.0  200.0000  400.00   674.75  1386.87500
      ALOPEZB                200.0  200.0000  200.00   200.00   200.00000
      CONVHANA_MM5           800.0  832.5000  865.00  1332.50  1800.00000
      GILRODRIGUEZ           150.0  175.0000  200.00   335.00   470.00000
      GSOTOR                  50.0  120.0000  120.00   170.00   245.00000
      RHOLTEN                 30.0  150.1875  189.25   317.50   568.46875


                         IQR
      user
      ALEINSUMOS    474.7500
```

```
ALOPEZB           0.0000
CONVHANA_MM5    500.0000
GILRODRIGUEZ    160.0000
GSOTOR           50.0000
RHOLTEN         167.3125
```

[91]:

```
do_boxplot(most_used_paints_df, x='user', y='expected_yield',␣
 ↪title='Rendimiento esperado (std) por usuario',
           rotate=True)
per_user_described_df.loc[:, ['expected_yield']]
```



[91]:

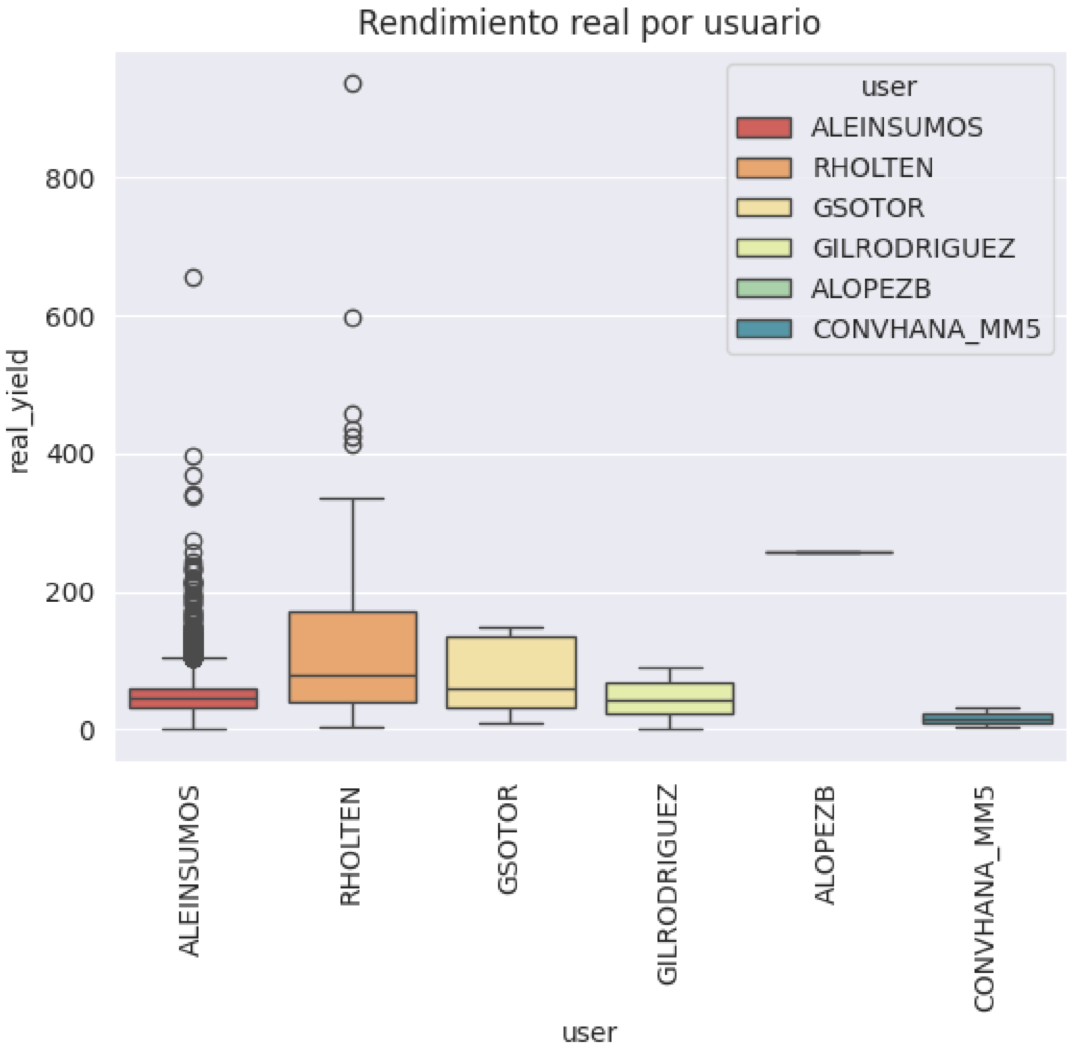
```
             expected_yield                                                \
             bottom_whisker        25%        50%        75% top_whisker
user
```

```
ALEINSUMOS        18.673219  44.440945  54.073034  67.637795   68.953881
ALOPEZB           54.073034  54.073034  54.073034  54.073034   54.073034
CONVHANA_MM5      54.073034  54.073034  54.073034  56.170375   58.267717
GILRODRIGUEZ      54.073034  60.855414  67.637795  67.637795   67.637795
GSOTOR            44.291339  44.291339  58.267717  67.637795   68.953881
RHOLTEN           18.673219  44.440945  54.201871  67.637795   68.953881
```

```
                    IQR
user
ALEINSUMOS    23.196850
ALOPEZB        0.000000
CONVHANA_MM5   2.097341
GILRODRIGUEZ   6.782381
GSOTOR        23.346457
RHOLTEN       23.196850
```

[92]:

```
do_boxplot(most_used_paints_df, x='user', y='real_yield', title='Rendimiento␣
 ↪real por usuario', rotate=True)
per_user_described_df.loc[:, ['real_yield']]
```

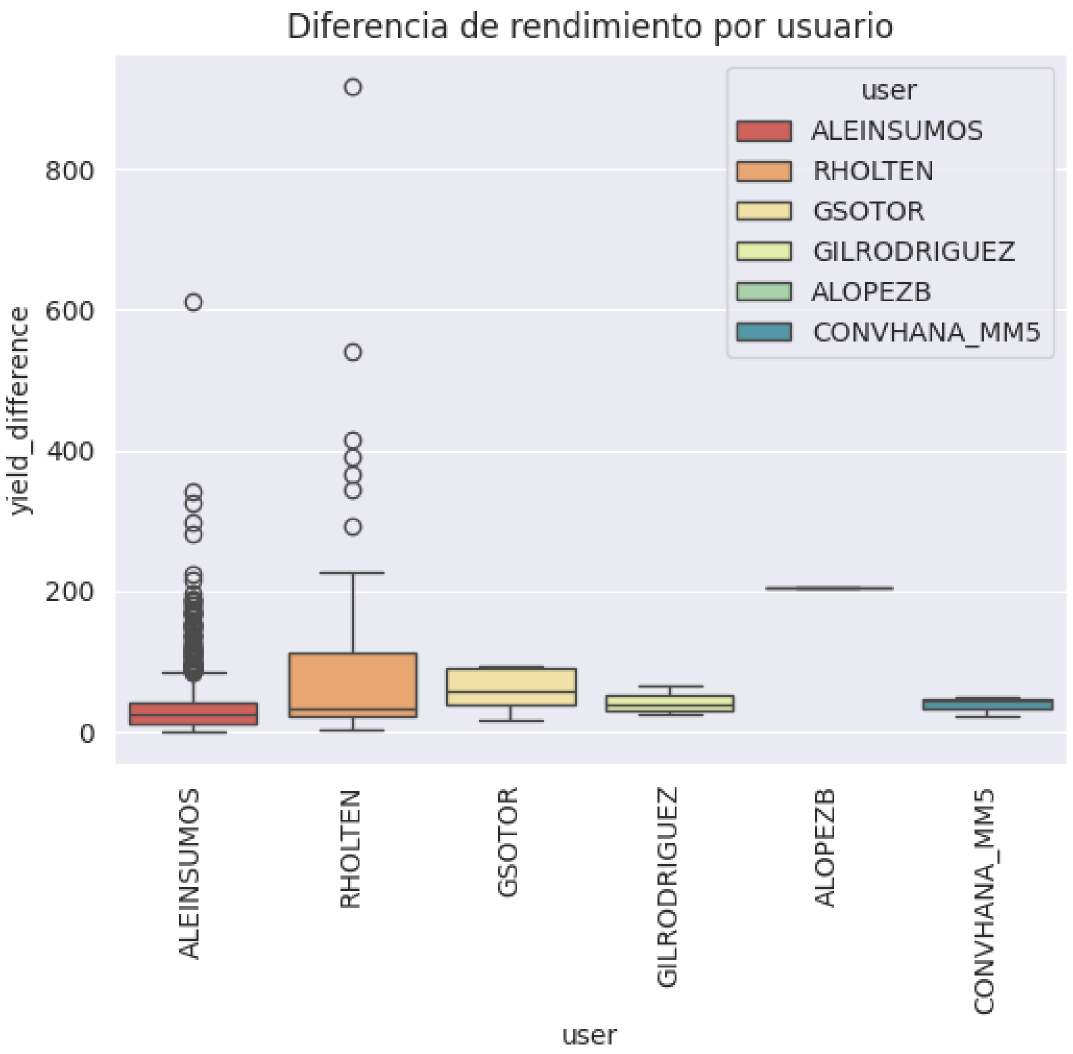


[92]:

```
                 real_yield                                                  \
             bottom_whisker         25%         50%         75% top_whisker
user
ALEINSUMOS         1.203259   32.233730   44.250772   60.538011  102.994434
ALOPEZB          257.713280  257.713280  257.713280  257.713280  257.713280
CONVHANA_MM5       3.802183    9.583781   15.365380   23.875948   32.386515
GILRODRIGUEZ       1.866800   22.660183   43.453567   67.320118   91.186670
GSOTOR             9.452911   30.464383   60.122118  135.830992  147.769935
RHOLTEN            4.479238   39.172452   78.686832  171.842777  370.848265


                     IQR
user
ALEINSUMOS     28.304281
```

```
ALOPEZB          0.000000
CONVHANA_MM5    14.292166
GILRODRIGUEZ    44.659935
GSOTOR         105.366609
RHOLTEN        132.670325
```

```
[93]: do_boxplot(most_used_paints_df, x='user', y='yield_difference',␣
       ↪title='Diferencia de rendimiento por usuario',
                 rotate=True)
      per_user_described_df.loc[:, ['yield_difference']]
```



```
[93]:            yield_difference                                              \
                   bottom_whisker         25%         50%          75% top_whisker
      user
```

```
ALEINSUMOS           0.006838    9.661765   24.411222   39.838202   85.102857
ALOPEZB            203.640246  203.640246  203.640246  203.640246  203.640246
CONVHANA_MM5        21.686519   32.294428   42.902337   46.586594   50.270851
GILRODRIGUEZ        24.184229   30.648932   37.113636   51.442316   65.770995
GSOTOR              15.830779   38.489497   58.184885   89.502218   91.539654
RHOLTEN              2.354336   21.023755   33.342580  110.884021  245.674419


                    IQR
user
ALEINSUMOS    30.176437
ALOPEZB        0.000000
CONVHANA_MM5  14.292166
GILRODRIGUEZ  20.793383
GSOTOR        51.012721
RHOLTEN       89.860266
```