```
1 from google.colab import drive
2 drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

```
1 # Importing the necessary libraries
2 import time
3
4 import numpy as np
5 import pandas as pd
6
7 import seaborn as sns
8 import matplotlib.pyplot as plt
```

Jaime

```
1 df = pd.read_csv("/content/gdrive/MyDrive/df_NUEVO.csv").copy()
```

```
1 print(df.head())
```

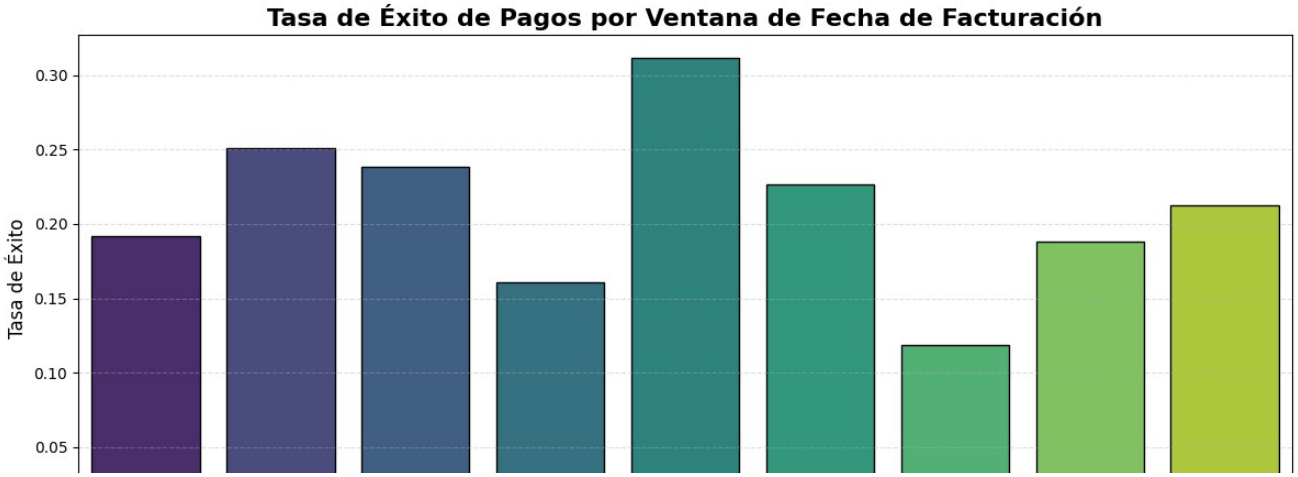
```
   idListaCobro  idCredito  consecutivoCobro  idBanco_x  montoExigible  \
0        155938     738973          41396434          2         622.87
1        155938     739017          41396435          2        1069.11
2        155939     739185          41396436          2        4340.83
3        155940     732324          41396437          2        2134.21
4        155940     737028          41396438          2         815.76

   montoCobrar  montoCobrado fechaCobroBanco  idRespuestaBanco  caso_exitoso  \
0       622.87          0.00             NaN               4.0             0
1      1069.11          0.00             NaN               4.0             0
2      4340.83       4340.83      02/01/2025               0.0             1
3      2134.21          0.00             NaN               4.0             0
4       815.76          0.00             NaN               4.0             0

   ... id_estrategia  servicio tiempo_feedback  Hora_ inicio  Hora_fin  \
0  ...             9         1            1800      08:00:00  14:59:00
1  ...             9         1            1800      08:00:00  14:59:00
2  ...             9         1            1800      08:00:00  14:59:00
3  ...             9         1            1800      08:00:00  14:59:00
4  ...             9         1            1800      08:00:00  14:59:00

      pagare  capital fechaAperturaCredito CobroExito  CobroDevuelta
0   14948.88   7848.0           19/12/2024       1.75           1.75
1   51317.28  18265.0           20/12/2024       1.75           1.75
2  104179.92  37080.0           21/12/2024       1.75           1.75
3  110978.92  39500.0           28/10/2024       1.75           1.75
4   42419.52  15098.0           02/12/2024       1.75           1.75
```

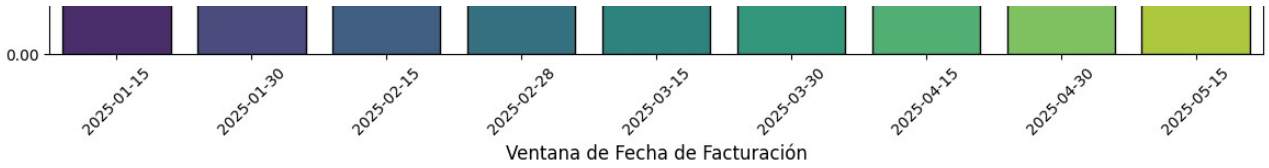
```
[5 rows x 24 columns]
```

```
# prompt: from the df dataframe, I want to get the distribution of payments. The paym

import pandas as pd
import matplotlib.pyplot as plt
df['fechaEnvioCobro'] = pd.to_datetime(df['fechaEnvioCobro'], dayfirst=True, errors='

# Define the payment date windows (e.g., 13th to 17th and 28th to end of month)
def assign_billing_date(date):
    if pd.isna(date):
        return None
    day = date.day
    # Window for 15th (covers 13th to 17th)
    if 13 <= day <= 17:
        return date.replace(day=15).date()
    # Window for 30th (covers 28th to end of month)
    elif 28 <= day:
         # Ensure the day does not exceed the number of days in the month
         last_day_of_month = date.days_in_month
         if day <= last_day_of_month:
            return date.replace(day=30).date() if last_day_of_month >= 30 else date.r
         else: # Should not happen with the 28 <= day condition if date is valid
             return None
    return None

df['billing_date'] = df['fechaEnvioCobro'].apply(assign_billing_date)

# Filter out rows where billing_date could not be assigned
df_billed = df.dropna(subset=['billing_date']).copy()

# Aggregate by idCredito and billing_date
# Get the max of 'caso_exitoso' to check if at least one payment attempt was successf
payment_distribution = df_billed.groupby(['idCredito', 'billing_date']).agg(
    payment_attempted=('idCredito', 'size'), # Count the number of attempts within th
    payment_successful=('caso_exitoso', 'max') # Check if any attempt was successful
).reset_index()

print("Distribución de Pagos por idCredito y Ventana de Fecha de Facturación:")
print(payment_distribution.head())

# Optional: Aggregate further to see overall success rate per billing date window
billing_date_summary = payment_distribution.groupby('billing_date').agg(
    total_credits=('idCredito', 'nunique'), # Count unique credits that had an attemp
    successful_credits=('payment_successful', lambda x: (x > 0).sum()) # Count credit
).reset_index()

# Calculate success rate
billing_date_summary['success_rate'] = billing_date_summary['successful_credits'] / b
```

```
48
49 print("\nResumen por Ventana de Fecha de Facturación:")
50 print(billing_date_summary)
51
52 # Optional: Visualize the success rate per billing date window
53 plt.figure(figsize=(12, 6))
54 sns.barplot(data=billing_date_summary, x='billing_date', y='success_rate', palette='v
55 plt.title('Tasa de Éxito de Pagos por Ventana de Fecha de Facturación', fontsize=16,
56 plt.xlabel('Ventana de Fecha de Facturación', fontsize=12)
57 plt.ylabel('Tasa de Éxito', fontsize=12)
58 plt.xticks(rotation=45)
59 plt.grid(axis='y', linestyle='--', alpha=0.4)
60 plt.tight_layout()
61 plt.show()
62
```

```
Distribución de Pagos por idCredito y Ventana de Fecha de Facturación:
   idCredito billing_date  payment_attempted  payment_successful
0       9872   2025-01-15                 14                   0
1      30466   2025-01-15                 14                   0
2      38430   2025-01-15                  3                   0
3      38430   2025-01-30                  4                   0
4      41330   2025-01-15                 14                   0

Resumen por Ventana de Fecha de Facturación:
  billing_date  total_credits  successful_credits  success_rate
0   2025-01-15           6143                1177      0.191600
1   2025-01-30           5997                1505      0.250959
2   2025-02-15           3238                 771      0.238110
3   2025-02-28           3134                 503      0.160498
4   2025-03-15           6271                1953      0.311434
5   2025-03-30           4792                1087      0.226836
6   2025-04-15           4075                 484      0.118773
7   2025-04-30           7531                1417      0.188156
8   2025-05-15           6837                1454      0.212666
<ipython-input-15-916fd98bf921>:54: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

  sns.barplot(data=billing_date_summary, x='billing_date', y='success_rate', palette=
```

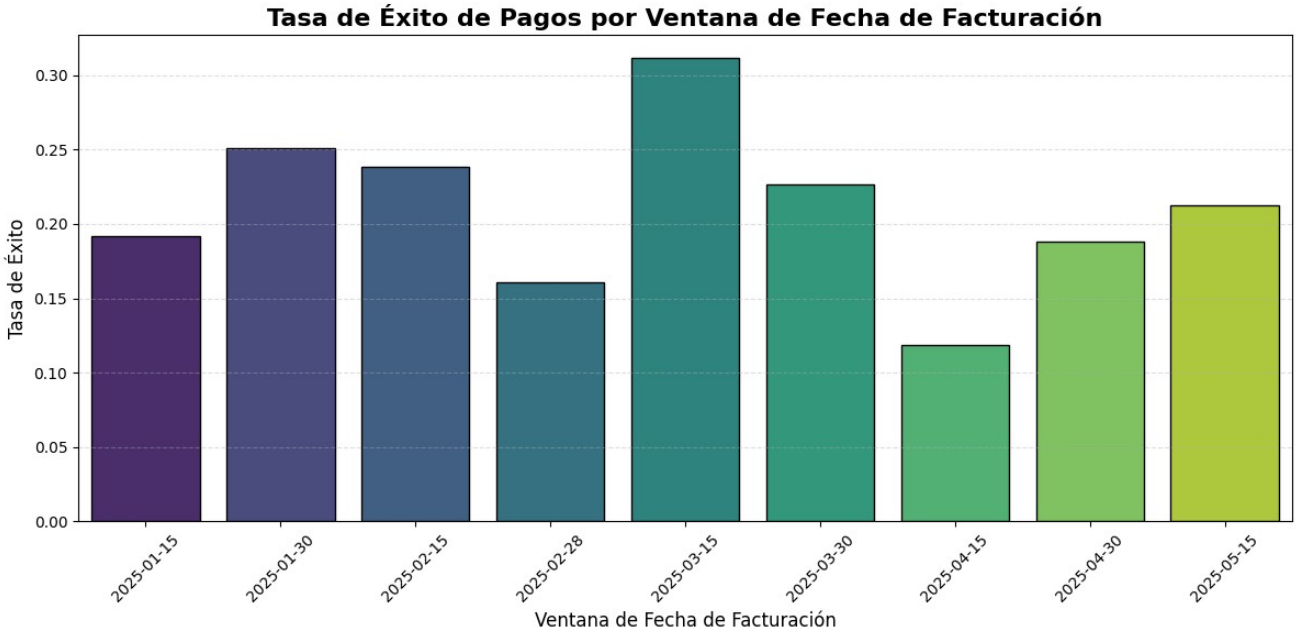




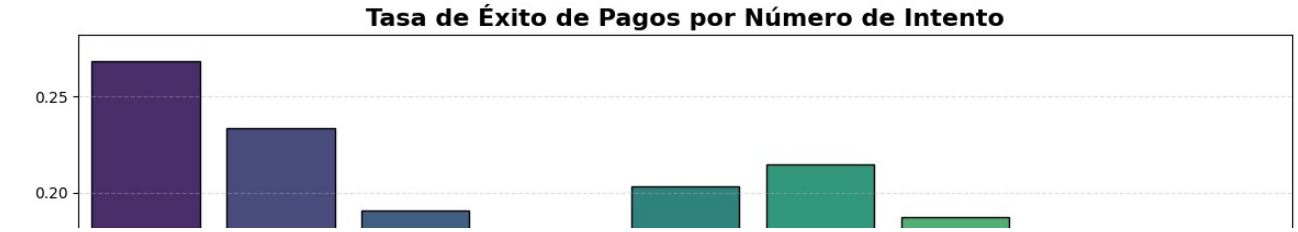

```
# prompt: from the df dataframe, I want to get the distribution of payments. The paym

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df['fechaEnvioCobro'] = pd.to_datetime(df['fechaEnvioCobro'], dayfirst=True, errors='

# Define the payment date windows (e.g., 13th to 17th and 28th to end of month)
def assign_billing_date(date):
    if pd.isna(date):
        return None
    day = date.day
    # Window for 15th (covers 13th to 17th)
    if 13 <= day <= 17:
        return date.replace(day=15).date()
    # Window for 30th (covers 28th to end of month)
    elif 28 <= day:
         # Ensure the day does not exceed the number of days in the month
         last_day_of_month = date.days_in_month
         if day <= last_day_of_month:
            return date.replace(day=30).date() if last_day_of_month >= 30 else date.r
         else: # Should not happen with the 28 <= day condition if date is valid
             return None
    return None

df['billing_date'] = df['fechaEnvioCobro'].apply(assign_billing_date)

# Filter out rows where billing_date could not be assigned
df_billed = df.dropna(subset=['billing_date']).copy()
```

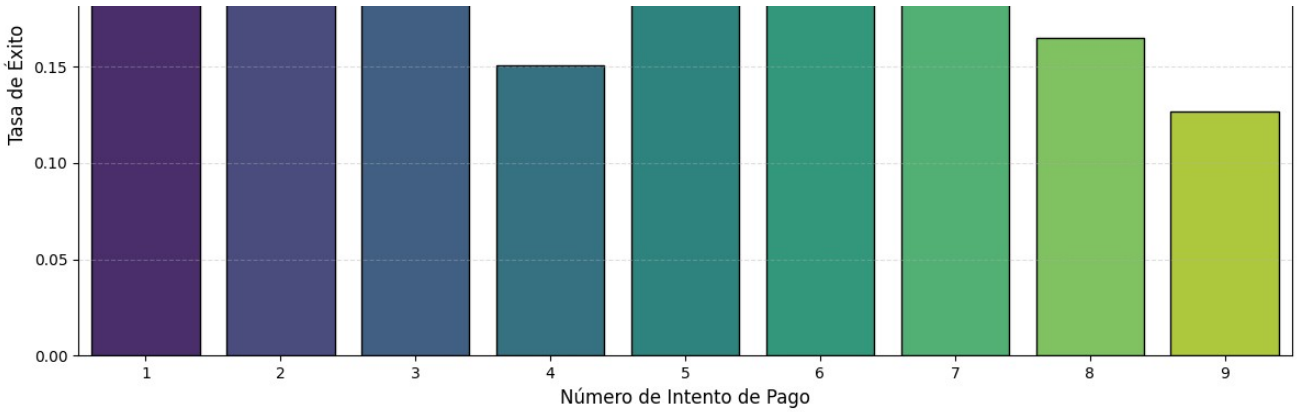
```
30
31 # Aggregate by idCredito and billing_date
32 # Get the max of 'caso_exitoso' to check if at least one payment attempt was successf
33 payment_distribution = df_billed.groupby(['idCredito', 'billing_date']).agg(
34     payment_attempted=('idCredito', 'size'), # Count the number of attempts within th
35     payment_successful=('caso_exitoso', 'max') # Check if any attempt was successful
36 ).reset_index()
37
38 print("Distribución de Pagos por idCredito y Ventana de Fecha de Facturación:")
39 print(payment_distribution.head())
40
41 # Optional: Aggregate further to see overall success rate per billing date window
42 billing_date_summary = payment_distribution.groupby('billing_date').agg(
43     total_credits=('idCredito', 'nunique'), # Count unique credits that had an attemp
44     successful_credits=('payment_successful', lambda x: (x > 0).sum()) # Count credit
45 ).reset_index()
46
47 payment_distribution['payment_number'] = payment_distribution.sort_values(by=['idCred
48
49 # Calculate success rate
50 billing_date_summary['success_rate'] = billing_date_summary['successful_credits'] / b
51
52 print("\nResumen por Ventana de Fecha de Facturación:")
53 print(billing_date_summary)
54
55 # Optional: Visualize the success rate per billing date window
56 plt.figure(figsize=(12, 6))
57 sns.barplot(data=billing_date_summary, x='billing_date', y='success_rate', palette='v
58 plt.title('Tasa de Éxito de Pagos por Ventana de Fecha de Facturación', fontsize=16,
59 plt.xlabel('Ventana de Fecha de Facturación', fontsize=12)
60 plt.ylabel('Tasa de Éxito', fontsize=12)
61 plt.xticks(rotation=45)
62 plt.grid(axis='y', linestyle='--', alpha=0.4)
63 plt.tight_layout()
64 plt.show()
65
66 # Calculate the success rate for each payment number
67 payment_number_summary = payment_distribution.groupby('payment_number')['payment_succ
68 # Create the bar plot
69 plt.figure(figsize=(12, 6))
70 sns.barplot(data=payment_number_summary, x='payment_number', y='payment_successful',
71 # Add aesthetics
72 plt.title('Tasa de Éxito de Pagos por Número de Intento', fontsize=16, fontweight='bo
73 plt.xlabel('Número de Intento de Pago', fontsize=12)
74 plt.ylabel('Tasa de Éxito', fontsize=12)
75 plt.xticks(rotation=0) # Keep x-axis labels horizontal
76 plt.grid(axis='y', linestyle='--', alpha=0.4) # Add a horizontal grid
77 plt.tight_layout() # Adjust layout to prevent labels from overlapping
78 plt.show() # Display the plot
79
```

```
Distribución de Pagos por idCredito y Ventana de Fecha de Facturación:
   idCredito billing_date  payment_attempted  payment_successful
0       9872   2025-01-15                 14                   0
1      30466   2025-01-15                 14                   0
2      38430   2025-01-15                  3                   0
3      38430   2025-01-30                  4                   0
4      41330   2025-01-15                 14                   0

Resumen por Ventana de Fecha de Facturación:
  billing_date  total_credits  successful_credits  success_rate
0   2025-01-15           6143                1177      0.191600
1   2025-01-30           5997                1505      0.250959
2   2025-02-15           3238                 771      0.238110
3   2025-02-28           3134                 503      0.160498
4   2025-03-15           6271                1953      0.311434
5   2025-03-30           4792                1087      0.226836
6   2025-04-15           4075                 484      0.118773
7   2025-04-30           7531                1417      0.188156
8   2025-05-15           6837                1454      0.212666
<ipython-input-20-137f5ccf6d2e>:57: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

  sns.barplot(data=billing_date_summary, x='billing_date', y='success_rate', palette=
```




```
<ipython-input-20-137f5ccf6d2e>:70: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

  sns.barplot(data=payment_number_summary, x='payment_number', y='payment_successful'
```





Double-click (or enter) to edit