

Universidad Nacional Autónoma de México

Facultad de Ciencias

Riesgo Tecnológico 2025-1

Laboratorio: Histogramas

Profesora:

Selene Marisol Martínez Ramírez

Ayudantes:

César Eduardo Jardines Mendoza

Itzel Azucena Delgado Díaz

Luis Angel Rojas Espinoza

Luis Rey Rutiaga Robles

INTEGRANTES:

Deloya Andrade Ana Valeria

Cortés Jiménez Carlos Daniel

Cruz Gonzalez Irvin Javier

Castro Reyes Angel

Cruz Blanco Gabriela

5 de octubre de 2024

1. Histogramas

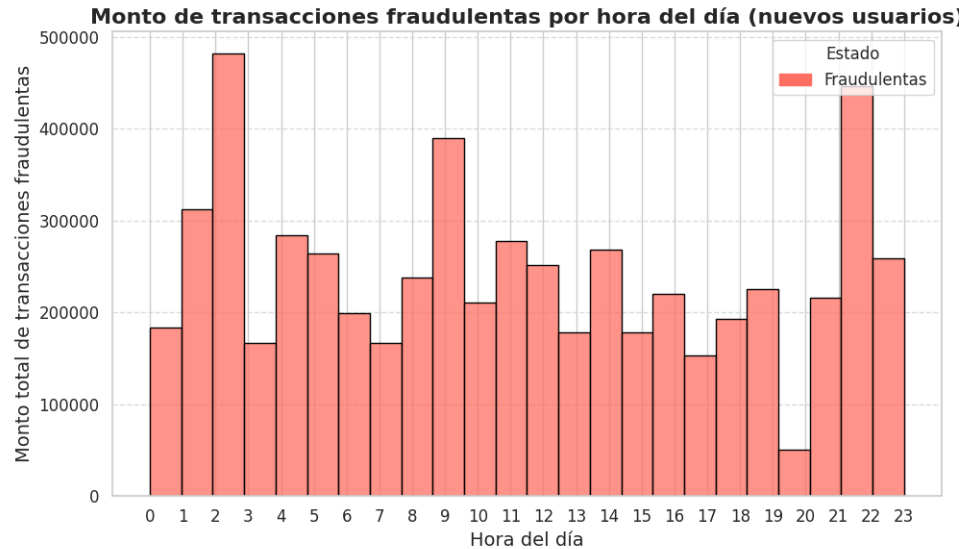
- Histograma del monto de las transacciones por hora del día con estado fraudolento de nuevos usuarios (Cuanto dinero históricamente se ha perdido más en las 24 horas del día):

A continuación se muestra la implementación de este histograma, para ejecutarlo basta con colocar en terminal el comando `python3 histograma1.py`:

```
1      import pandas as pd
2      import matplotlib.pyplot as plt
3      import seaborn as sns
4      import matplotlib.patches as mpatches
5
6      # Cargar el archivo CSV
7      data = pd.read_csv('transacciones.csv')
8
9      # Crear una nueva columna combinando 'date' y 'time'
10     data['transaction_time'] = pd.to_datetime(data['date'] + ' ' + data['time']
11     ], format='%d/%m/%Y %H:%M')
12
13     # Filtrar para obtener solo transacciones fraudulentas de nuevos usuarios
14     fraudulent_transactions_new_users = data.loc[(data['status'] == 'fraudulent
15     ') & (data['new_user'] == True)].copy()
16
17     # Extraer la hora del día de la columna 'transaction_time'
18     fraudulent_transactions_new_users.loc[:, 'hour'] =
19     fraudulent_transactions_new_users['transaction_time'].dt.hour
20
21     # Añadimos un estilo a la gráfica
22     sns.set(style="whitegrid")
23
24     # Se define y se crea la gráfica
25     plt.figure(figsize=(10, 6))
26     colors = ["#FF6F61"]
27
28     # Crear el histograma con el monto por hora
29     sns.histplot(
30         data=fraudulent_transactions_new_users,
31         x='hour',
32         weights='amount', # Aquí el peso será el monto de la transacción
33         bins=24, # Una barra por cada hora del día
34         color=colors[0],
35         edgecolor='black'
36     )
37
38     # Títulos y etiquetas
39     plt.title('Monto de transacciones fraudulentas por hora del día (nuevos
40     usuarios)', fontsize=16, fontweight='bold')
41     plt.xlabel('Hora del día', fontsize=14)
42     plt.ylabel('Monto total de transacciones fraudulentas', fontsize=14)
43
44     # Personalizamos los ejes
45     plt.xticks(range(0, 24), fontsize=12)
46     plt.yticks(fontsize=12)
47
48     # Cuadrícula en la gráfica
49     plt.grid(axis='y', linestyle='--', alpha=0.7)
50
51     # Crear la leyenda "Fraudulentas"
52     fraud_patch = mpatches.Patch(color='#FF6F61', label='Fraudulentas')
53
54     # Guardar la gráfica como imagen en lugar de mostrarla
55     plt.legend(handles=[fraud_patch], title='Estado', loc='upper right',
56     fontsize=12)
57     plt.tight_layout()
58
59     # Guardar la gráfica en un archivo PNG
60     plt.savefig('IMAGE/histograma1.png')
61
62     plt.show()
```

Listing 1: Implementación del Histograma del monto de transacciones por hora del día

El histograma es el siguiente:



Este histograma permite visualizar las horas en las que se realizan más transacciones fraudulentas en términos de monto, lo cual puede proporcionar información valiosa para ajustar las medidas de seguridad y monitorear con más detalle ciertas horas críticas del día.

En términos de análisis de riesgos, si el histograma muestra un pico durante ciertas horas (por ejemplo, entre la medianoche y las 3:00 a.m.), puede indicar que los atacantes prefieren operar en esos momentos, lo que lleva a la implementación de controles adicionales durante esas horas.

- **Histograma de la distribución de nuevos usuarios con una transacción fraudolenta (Cuantos usuarios nuevos tuvieron una transacción fraudolenta vs los usuarios que no son nuevos):**

A continuación se muestra la implementación de este histograma, para ejecutarlo basta con colocar en terminal el comando `python3 histograma2.py`:

```

1  import pandas as pd
2  import matplotlib.pyplot as plt
3  import seaborn as sns
4  import matplotlib.patches as mpatches
5
6  # Cargar el CSV
7  data = pd.read_csv('transacciones.csv')
8
9  # Filtra para obtener solo transacciones fraudulentas
10 fraudulent_transactions = data[data['status'] == 'fraudulent']
11
12 # Formato para extraer la información para las columnas de los usuarios no
nuevos y nuevos
13 fraudulent_transactions_users = fraudulent_transactions[
14     (fraudulent_transactions['new_user'] == True) |
15     (fraudulent_transactions['new_user'] == False)]
16
17 # Se añade un estilo a la gráfica
18 sns.set(style="whitegrid")
19
20 # Se define y se crea la gráfica
21 plt.figure(figsize=(10, 6))
22 colors = ["#73a13f", "#73a13f"]
23

```

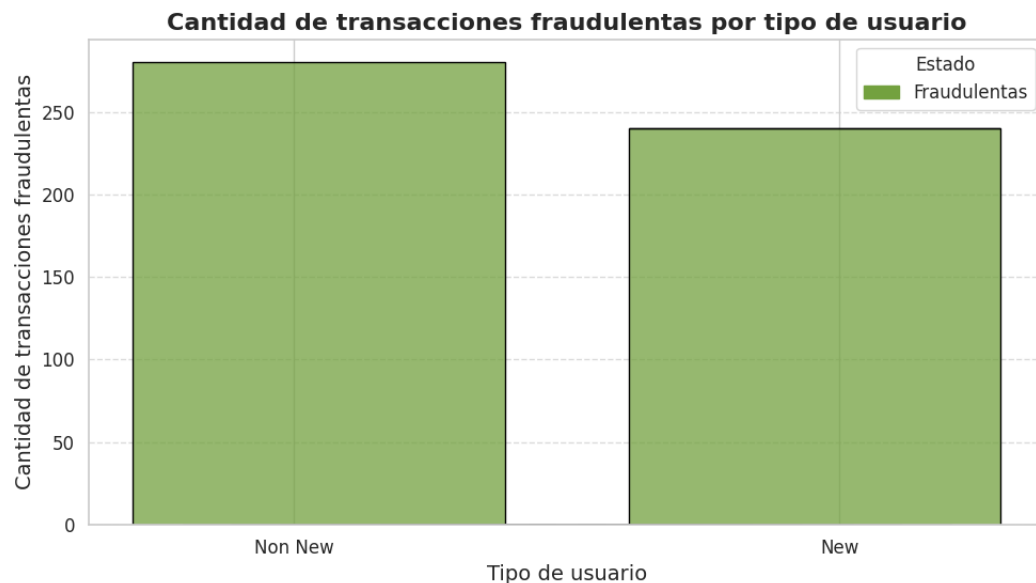
```

24     # Personalizamos las barras
25     sns.histplot(
26         data=fraudulent_transactions_users,
27         x='new_user',
28         hue='new_user',
29         multiple='stack',
30         palette=colors,
31         shrink=7.5,
32         edgecolor='black')
33
34     # Titulos
35     plt.title('Cantidad de transacciones fraudulentas por tipo de usuario',
36             fontsize=16, fontweight='bold')
37     plt.xlabel('Tipo de usuario', fontsize=14)
38     plt.ylabel('Cantidad de transacciones fraudulentas', fontsize=14)
39
40     # Personalizamos los ejes
41     plt.xticks([0, 1], ['Non New', 'New'], fontsize=12)
42     plt.yticks(fontsize=12)
43
44     # Cuadrícula en la gráfica
45     plt.grid(axis='y', linestyle='--', alpha=0.7)
46
47     # Crear la leyenda "Fraudulentas" de Estado
48     new_patch = mpatches.Patch(color='#73a13f', label='Fraudulentas')
49
50     # Muestra la gráfica
51     plt.legend(handles=[new_patch], title='Estado', loc='upper right', fontsize=12)
52     plt.tight_layout()
53     plt.show()

```

Listing 2: Implementación de histograma de la distribución de nuevos usuarios con una transacción fraudolenta.

El histograma es el siguiente:



Si bien hay una cantidad significativa de transacciones fraudulentas realizadas por los usuarios nuevos, existe una mayor cantidad realizada por aquellos usuarios que no son nuevos. Esto puede deberse a varios factores como que este sector de los usuarios tiene más experiencia y conocimiento en el manejo del sistema.

Con la información recabada en este histograma puede tomarse medidas como la implementación de un mayor monitoreo y análisis del comportamiento en usuarios que ya llevan tiempo utilizando el servicio

cuando estos llevan a cabo transacciones. Utilizando así los datos recabados para reforzar la seguridad del sistema de manera eficiente.

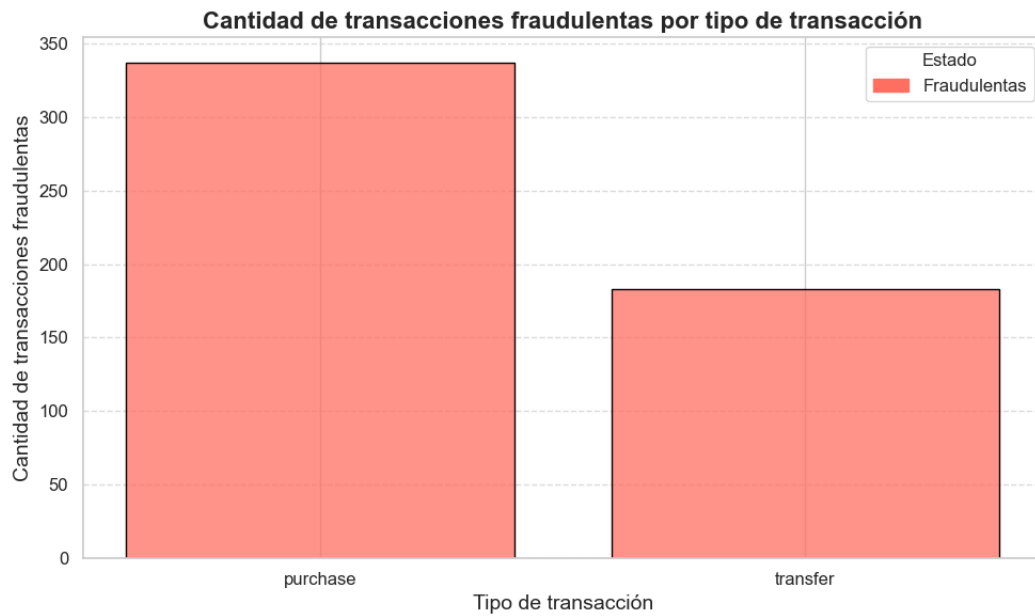
- **Histograma del tipo de transacción y el estado de la transacción fraudolenta (Cuántos estados de transacciones fraudolentas tuvieron las transacciones purchase y transfer):**

El código usado fue el siguiente:

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4 import matplotlib.patches as mpatches
5
6 # Cargar el CSV
7 data = pd.read_csv('transacciones.csv')
8
9 # Filtrar para obtener solo transacciones fraudolentas
10 fraudulent_transactions = data[data['status'] == 'fraudulent']
11
12 # Formato para extraer solo la compra y transferencia de la columna 'purchase' y '
13 transfer'
14 fraudulent_transactions_filtered = fraudulent_transactions[
15     (fraudulent_transactions['transaction_type'] == 'purchase') |
16     (fraudulent_transactions['transaction_type'] == 'transfer')
17 ]
18
19 # Añadimos un estilo a la gráfica
20 sns.set(style="whitegrid")
21
22 # Se define y se crea la gráfica
23 plt.figure(figsize=(10, 6))
24 colors = ["#FF6F61", "#6B5B95"]
25
26 # Personalizamos las barras
27 sns.histplot(
28     data=fraudulent_transactions_filtered,
29     x='transaction_type',
30     hue='status',
31     multiple='stack',
32     palette=colors,
33     shrink=0.8,
34     edgecolor='black'
35 )
36
37 # Titulos
38 plt.title('Cantidad de transacciones fraudolentas por tipo de transacción',
39     fontsize=16, fontweight='bold')
40 plt.xlabel('Tipo de transacción', fontsize=14)
41 plt.ylabel('Cantidad de transacciones fraudolentas', fontsize=14)
42
43 # Personalizamos los ejes
44 plt.xticks(fontsize=12)
45 plt.yticks(fontsize=12)
46
47 # Cuadrícula en la gráfica
48 plt.grid(axis='y', linestyle='--', alpha=0.7)
49
50 # Crear la leyenda "Fraudolentas"
51 fraud_patch = mpatches.Patch(color='#FF6F61', label='Fraudolentas')
52
53 # Mostramos la gráfica
54 plt.legend(handles=[fraud_patch], title='Estado', loc='upper right', fontsize=12)
55 plt.tight_layout()
56 plt.show()
```

Listing 3: Implementación histograma del tipo de transacción y el estado de la transacción fraudolenta.

También se adjuntara el archivo .py, abrimos nuestra terminal y ejecutamos **python3 histograma3.py** y la gráfica resultante es la siguiente:



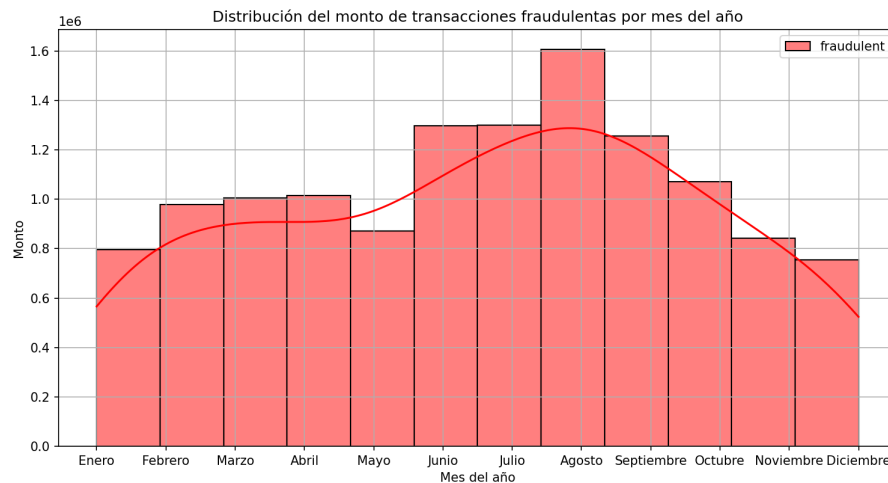
Podemos ver que la cantidad de transacciones de tipo "purchase" es más alta que las de "transfer", esto nos puede decir que los sistemas de compra son más propensos a que sean atacados, siendo que podría haber vulnerabilidades en la seguridad al realizar una compra y así haber menos control en esta área en comparación con las transferencias, por lo que los estafadores prefieren realizar fraudes mediante compras ya que les sería más sencillo el poder realizar una acción fraudolenta entre las transacciones legítimas, ya que si realizan el fraude en transferencias esto puede ser más fácil de rastrear.

■ Histograma de los meses con transacciones fraudulentas:

A continuación se muestra la implementación de este histograma, para ejecutarlo basta con colocar en terminal el comando **python3 histograma5.py**:

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4
5 # Cargar los datos desde el archivo CSV
6 data = pd.read_csv('transacciones.csv')
7
8 # Filtrar para obtener solo transacciones fraudulentas
9 fraudulent_transactions = data[(data['status'] == 'fraudulent')]
10
11 # Formato para extraer solo el mes de la columna 'date'
12 fraudulent_transactions['month'] = pd.to_datetime(fraudulent_transactions['date'], format='%d/%m/%Y').dt.month
13
14 # Crear una lista de nombres de los meses
15 months_names = ['Enero', 'Febrero', 'Marzo', 'Abril', 'Mayo', 'Junio', 'Julio', 'Agosto', 'Septiembre', 'Octubre', 'Noviembre', 'Diciembre']
16
17 # Se define y se crea la gráfica con pesos
18 plt.figure(figsize=(12, 6)) # Ajusta el tamaño de la figura
19 sns.histplot(
20     data=fraudulent_transactions, x='month', weights='amount', bins=12, kde=True,
21     color='red', label='fraudulent')
22
23 # Cambiar las etiquetas del eje X a los nombres de los meses
24 plt.xticks(ticks=range(1, 13), labels=months_names)
25
26 # Títulos
27 plt.title('Distribución del monto de transacciones fraudulentas por mes del año')
28
29 # Etiquetas de los ejes
30 plt.xlabel('Mes del año')
31 plt.ylabel('Monto')
32 plt.grid(True)
33
34 # Mostramos la gráfica
35 plt.legend()
36 plt.show()
```

Listing 4: Implementación histograma de los meses con transacciones fraudulentas.



De acuerdo al histograma se puede indicar que los fraudes son más comunes durante la segunda mitad del año, especialmente entre julio y septiembre, lo que podría coincidir con eventos estacionales o períodos

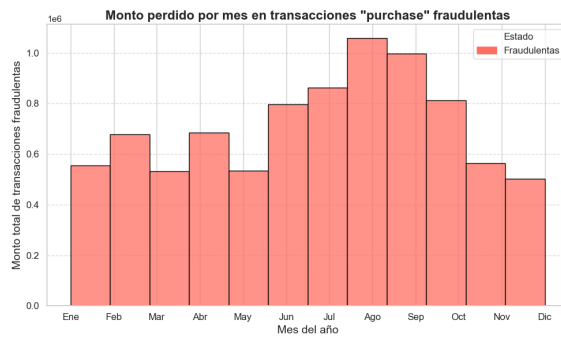
específicos de mayor actividad económica o comercial. Después del máximo, los fraudes disminuyen progresivamente hacia diciembre.

- **Histograma del monto perdido por mes por transacciones de tipo purchase con estado fraudolenta:**

A continuación se muestra la implementación de este histograma, para que se ejecute vamos a colocar en terminal el comando **python3 histograma6.py**:

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3  import seaborn as sns
4  import matplotlib.patches as mpatches
5
6  # Cargar el archivo CSV
7  data = pd.read_csv('transacciones.csv')
8
9  # Crear una nueva columna combinando 'date' y 'time'
10 data['transaction_time'] = pd.to_datetime(data['date'] + ' ' + data['time'], format
    = '%d/%m/%Y %H:%M')
11
12 # Filtrar transacciones fraudulentas de tipo purchase
13 fraudulent_purchase_transactions = data.loc[(data['status'] == 'fraudulent') &
14 (data['transaction_type'] == 'purchase'
    )].copy()
15
16 # Extraer el mes de la columna 'transaction_time'
17 fraudulent_purchase_transactions['month'] = fraudulent_purchase_transactions['
    transaction_time'].dt.month
18
19 # Añadimos un estilo a la gráfica
20 sns.set(style="whitegrid")
21
22 # Se define y se crea la gráfica
23 plt.figure(figsize=(10, 6))
24 colors = ["#FF6F61"]
25
26 # Crear el histograma con el monto por mes
27 sns.histplot(
28     data=fraudulent_purchase_transactions,
29     x='month',
30     weights='amount', # Aquí el peso será el monto de la transacción
31     bins=12, # Una barra por cada mes
32     color=colors[0],
33     edgecolor='black'
34 )
35
36 # Títulos y etiquetas
37 plt.title('Monto perdido por mes en transacciones "purchase" fraudulentas',
    fontsize=16, fontweight='bold')
38 plt.xlabel('Mes del año', fontsize=14)
39 plt.ylabel('Monto total de transacciones fraudulentas', fontsize=14)
40
41 # Personalizamos los ejes
42 plt.xticks(range(1, 13), ['Ene', 'Feb', 'Mar', 'Abr', 'May', 'Jun', 'Jul', 'Ago', '
    Sep', 'Oct', 'Nov', 'Dic'], fontsize=12)
43 plt.yticks(fontsize=12)
44
45 # Cuadrícula en la gráfica
46 plt.grid(axis='y', linestyle='--', alpha=0.7)
47
48 # Crear la leyenda "Fraudulentas"
49 fraud_patch = mpatches.Patch(color='#FF6F61', label='Fraudulentas')
50
51 # Guardar la gráfica como imagen en lugar de mostrarla
52 plt.legend(handles=[fraud_patch], title='Estado', loc='upper right', fontsize=12)
53 plt.tight_layout()
54
55 # Guardar la gráfica en un archivo PNG
56 plt.savefig('IMAGE/histograma6.png')
57
58 plt.show()
```

Listing 5: Implementación histograma del monto perdido por mes por transacciones de tipo purchase con estado fraudolenta.



Se observa que los meses de julio y agosto presentan el mayor número de fraudes, mientras que enero, noviembre y diciembre tienen los niveles más bajos. Entre enero y mayo, los montos se mantienen estables, pero a partir de junio se da un aumento significativo que culmina en agosto, seguido de una disminución en septiembre. Aunque las compras de fin de año suelen ser altas, noviembre y diciembre registran menos fraudes, lo que podría estar relacionado con medidas preventivas o cambios en el comportamiento de los consumidores.
