

# Projeto 01



No dinâmico setor de e-commerce atual, a entrega rápida e eficiente de pedidos é crucial para o sucesso dos negócios.

Para garantir o cumprimento ininterrupto dos pedidos, as empresas frequentemente estabelecem parcerias com instituições terceirizadas de entrega para enviar seus produtos aos clientes.

No entanto, gerenciar as cobranças efetuadas por essas empresas pode ser desafiador, especialmente ao lidar com um grande volume de pedidos.

Esse é um dos problemas em tempo real enfrentados por empresas B2B, quando as cobranças estimadas para uma mesma fatura não correspondem.

Neste projeto, apresentaremos uma solução para um desses problemas, com foco na avaliação da exatidão das taxas cobradas pelas empresas terceirizadas na entrega de mercadorias em transações B2B.

O objetivo é garantir que as empresas contratantes sejam cobradas de forma adequada pelos serviços prestados pelas empresas de entrega.

## Estudo de Caso : Fraude em E-commerce B2B

A Empresa ABC opera uma plataforma de e-commerce e processa milhares de pedidos diariamente.

Para entregar esses pedidos, a ABC firmou parcerias com várias empresas de entregas na Índia, que cobram com base no peso dos produtos e na distância entre o armazém e o endereço de entrega do cliente.

A ABC deseja verificar se as taxas cobradas pelas empresas de entregas para cada pedido estão corretas.

A ABC possui dados internos divididos em três relatórios:

- **Relatório de Pedidos do Website;**
- **Lista Mestre de Produtos** "*Master SKU-Stock Keeping Unit*";
- **Mapeamento de CEPs dos Armazéns** para todos os códigos postais da Índia.

Além disso, eles recebem dados de faturamento das empresas de entrega.

O *relatório de pedidos do site* inclui IDs de pedidos e produtos (*SKUs*) para cada pedido.

O *lista mestre de produtos* fornece o peso bruto de cada produto, necessário para calcular o peso total de cada pedido.

As *faturas das empresas de entrega* contêm informações como número "*AWB - Air Waybill*", ID do pedido, peso da remessa, código postal de retirada do armazém, código postal de entrega do cliente, área de entrega, a taxa por remessa e o tipo de remessa.

A ABC quer comparar o peso total de cada pedido, calculado usando a *lista mestre de produtos*, com o peso informado pela empresa de entrega em sua fatura. O peso deve ser arredondado para o múltiplo mais próximo de 0,5 kg para determinar o peso da remessa.

O *mapeamento de CEPs dos armazéns* para todos os códigos postais da Índia é usado para determinar a área de entrega, que deve ser comparada com a área informada pela empresa de entrega.

Além disso, a ABC deve aplicar a lógica de cálculo das taxas com base no peso da remessa, área de entrega e tipo de remessa listados na fatura da empresa de entrega.

A tabela de tarifas de entrega fornece uma taxa fixa e uma taxa adicional para cada faixa de peso e CEP. A cobrança total por remessa deve ser calculada somando a taxa fixa e quaisquer cobranças adicionais com base no peso da remessa.

A solução final deve incluir uma tabela com as seguintes colunas:

| Order ID   | Total Weight | Weight Slab | Delivery Zone as per ABC | Expected Charge as per ABC (Rs.) | AWB Number | Total Weight as per Courier Company (KG) | Delivery Zone Charged by Courier Company | Billing Amount (Rs.) | Weight Slab Charged by Courier Company (KG) |
|------------|--------------|-------------|--------------------------|----------------------------------|------------|------------------------------------------|------------------------------------------|----------------------|---------------------------------------------|
| 2001827036 | ...          | ...         | ...                      | ...                              | ...        | ...                                      | ...                                      | ...                  | ...                                         |
| 2001821995 | ...          | ...         | ...                      | ...                              | ...        | ...                                      | ...                                      | ...                  | ...                                         |
| ...        | ...          | ...         | ...                      | ...                              | ...        | ...                                      | ...                                      | ...                  | ...                                         |

A empresa ABC também quer analisar a precisão das taxas cobradas pela empresa de entrega. Eles desejam uma tabela de resumo incluindo as seguintes informações:

| Description                                       | Count | Amount (Rs.) |
|---------------------------------------------------|-------|--------------|
| Total Orders where ABC has been correctly charged | ...   | ...          |
| Total Orders where ABC has been overcharged       | ...   | ...          |
| Total Orders where ABC has been undercharged      | ...   | ...          |

---

## Solução

### Carregando as Bibliotecas e Configurações

```
In [1]: import pandas as pd
import plotly.express as px
```

### Carregando os Dataset's

```
In [2]: #Relatório de Pedidos do Site
```

```
In [3]: order_report = pd.read_csv('Order Report.csv')
```

```
In [4]: #Lista Mestre de Produtos
```

```
In [5]: sku_master = pd.read_csv('SKU Master.csv')
```

```
In [6]: #Mapeamento de CEPs dos Armazéns
```

```
In [7]: pincode_mapping = pd.read_csv('pincodes.csv')
```



```
In [8]: #Faturas das Empresas de Entrega
```

```
In [9]: courier_invoice = pd.read_csv('Invoice.csv')
```

```
In [10]: #Tabela de Tarifas de Entrega
```

```
In [11]: courier_company_rates = pd.read_csv('Courier Company - Rates.csv')
```

```
In [ ]:
```

## Exibindo os Dataset's

```
In [12]: #Relatório de Pedidos do Site
```

```
In [13]: print("Order Report:")
print()
print(order_report.head())
```

Order Report:

|   | ExternOrderNo | SKU           | Order Qty | Unnamed: 3 | Unnamed: 4 |
|---|---------------|---------------|-----------|------------|------------|
| 0 | 2001827036    | 8904223818706 | 1.0       | NaN        | NaN        |
| 1 | 2001827036    | 8904223819093 | 1.0       | NaN        | NaN        |
| 2 | 2001827036    | 8904223819109 | 1.0       | NaN        | NaN        |
| 3 | 2001827036    | 8904223818430 | 1.0       | NaN        | NaN        |
| 4 | 2001827036    | 8904223819277 | 1.0       | NaN        | NaN        |

```
In [14]: #Lista Mestre de Produtos
```

```
In [15]: print("nSKU Master:")
print()
print(sku_master.head())
```

nSKU Master:

|   | SKU           | Weight (g) | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 |
|---|---------------|------------|------------|------------|------------|
| 0 | 8904223815682 | 210        | NaN        | NaN        | NaN        |
| 1 | 8904223815859 | 165        | NaN        | NaN        | NaN        |
| 2 | 8904223815866 | 113        | NaN        | NaN        | NaN        |
| 3 | 8904223815873 | 65         | NaN        | NaN        | NaN        |
| 4 | 8904223816214 | 120        | NaN        | NaN        | NaN        |

In [16]: *#Mapeamento de CEPs dos Armazéns*

```
In [17]: print("nPincode Mapping:")
print()
print(pinode_mapping.head())
```

nPincode Mapping:

|   | Warehouse | Pincode | Customer | Pincode | Zone | Unnamed: 3 | Unnamed: 4 |
|---|-----------|---------|----------|---------|------|------------|------------|
| 0 |           | 121003  |          | 507101  | d    | NaN        | NaN        |
| 1 |           | 121003  |          | 486886  | d    | NaN        | NaN        |
| 2 |           | 121003  |          | 532484  | d    | NaN        | NaN        |
| 3 |           | 121003  |          | 143001  | b    | NaN        | NaN        |
| 4 |           | 121003  |          | 515591  | d    | NaN        | NaN        |

In [18]: *#Faturas das Empresas de Entrega*

```
In [19]: print("nCourier Invoice:")
print()
print(courier_invoice.head())
```

nCourier Invoice:

|   | AWB Code      | Order ID   | Charged Weight | Warehouse Pincode | \ |
|---|---------------|------------|----------------|-------------------|---|
| 0 | 1091117222124 | 2001806232 | 1.30           | 121003            |   |
| 1 | 1091117222194 | 2001806273 | 1.00           | 121003            |   |
| 2 | 1091117222931 | 2001806408 | 2.50           | 121003            |   |
| 3 | 1091117223244 | 2001806458 | 1.00           | 121003            |   |
| 4 | 1091117229345 | 2001807012 | 0.15           | 121003            |   |

|   | Customer Pincode | Zone | Type of Shipment | Billing Amount (Rs.) |
|---|------------------|------|------------------|----------------------|
| 0 | 507101           | d    | Forward charges  | 135.0                |
| 1 | 486886           | d    | Forward charges  | 90.2                 |
| 2 | 532484           | d    | Forward charges  | 224.6                |
| 3 | 143001           | b    | Forward charges  | 61.3                 |
| 4 | 515591           | d    | Forward charges  | 45.4                 |

In [20]: *#Tabela de Tarifas de Entrega*

```
In [21]: print("\nCourier Company rates:")
print()
print(courier_company_rates.head())
```

Courier Company rates:

|   | fwd_a_fixed | fwd_a_additional | fwd_b_fixed | fwd_b_additional | fwd_c_fixed | \ |
|---|-------------|------------------|-------------|------------------|-------------|---|
| 0 | 29.5        | 23.6             | 33          | 28.3             | 40.1        |   |

|   | fwd_c_additional | fwd_d_fixed | fwd_d_additional | fwd_e_fixed | \ |
|---|------------------|-------------|------------------|-------------|---|
| 0 | 38.9             | 45.4        | 44.8             | 56.6        |   |

|   | fwd_e_additional | rto_a_fixed | rto_a_additional | rto_b_fixed | \ |
|---|------------------|-------------|------------------|-------------|---|
| 0 | 55.5             | 13.6        | 23.6             | 20.5        |   |

|   | rto_b_additional | rto_c_fixed | rto_c_additional | rto_d_fixed | \ |
|---|------------------|-------------|------------------|-------------|---|
| 0 | 28.3             | 31.9        | 38.9             | 41.3        |   |

|   | rto_d_additional | rto_e_fixed | rto_e_additional |
|---|------------------|-------------|------------------|
| 0 | 44.8             | 50.7        | 55.5             |

In [ ]:

## Verificando os Valores Ausentes

```
In [22]: #Relatório de Pedidos do Site
```

```
In [23]: print("Missing values in Website Order Report:")
print()
print(order_report.isnull().sum())
```

Missing values in Website Order Report:

|               |       |
|---------------|-------|
| ExternOrderNo | 0     |
| SKU           | 0     |
| Order Qty     | 0     |
| Unnamed: 3    | 400   |
| Unnamed: 4    | 400   |
| dtype:        | int64 |

```
In [24]: #Lista Mestre de Produtos
```

```
In [25]: print("Missing values in SKU Master:")
print()
print(sku_master.isnull().sum())
```

Missing values in SKU Master:

|            |       |
|------------|-------|
| SKU        | 0     |
| Weight (g) | 0     |
| Unnamed: 2 | 66    |
| Unnamed: 3 | 66    |
| Unnamed: 4 | 66    |
| dtype:     | int64 |

```
In [26]: #Mapeamento de CEPs dos Armazéns
```

```
In [27]: print("Missing values in Pincode Mapping:")
print()
print(pincode_mapping.isnull().sum())
```

Missing values in Pincode Mapping:

|                   |     |
|-------------------|-----|
| Warehouse Pincode | 0   |
| Customer Pincode  | 0   |
| Zone              | 0   |
| Unnamed: 3        | 124 |
| Unnamed: 4        | 124 |

dtype: int64

In [28]: *#Faturas das Empresas de Entrega*

```
In [29]: print("Missing values in Courier Invoice:")
print()
print(courier_invoice.isnull().sum())
```

Missing values in Courier Invoice:

|                      |   |
|----------------------|---|
| AWB Code             | 0 |
| Order ID             | 0 |
| Charged Weight       | 0 |
| Warehouse Pincode    | 0 |
| Customer Pincode     | 0 |
| Zone                 | 0 |
| Type of Shipment     | 0 |
| Billing Amount (Rs.) | 0 |

dtype: int64

In [30]: *#Tabela de Tarifas de Entrega*

```
In [31]: print("Missing values in courier company rates:")
print()
print(courier_company_rates.isnull().sum())
```



Missing values in courier company rates:

```
fwd_a_fixed      0
fwd_a_additional 0
fwd_b_fixed      0
fwd_b_additional 0
fwd_c_fixed      0
fwd_c_additional 0
fwd_d_fixed      0
fwd_d_additional 0
fwd_e_fixed      0
fwd_e_additional 0
rto_a_fixed      0
rto_a_additional 0
rto_b_fixed      0
rto_b_additional 0
rto_c_fixed      0
rto_c_additional 0
rto_d_fixed      0
rto_d_additional 0
rto_e_fixed      0
rto_e_additional 0
dtype: int64
```

In [ ]:

## Limpeza dos Dataset's

In [32]: *#Relatório de Pedidos do Site*

```
In [33]: order_report_1 = order_report[['ExternOrderNo', 'SKU', 'Order Qty']]
print(order_report_1.isnull().sum())
```

```
ExternOrderNo    0
SKU              0
Order Qty        0
dtype: int64
```

In [34]: *#Lista Mestre de Produtos*

```
In [35]: sku_master_1 = sku_master[['SKU', 'Weight (g)']]
print(sku_master_1.isnull().sum())
```

```
SKU          0
Weight (g)   0
dtype: int64
```

```
In [36]: #Mapeamento de CEPs dos Armazéns
```

```
In [37]: pincode_mapping_1 = pincode_mapping[['Warehouse Pincode', 'Customer Pincode', 'Zone']]
print(pincode_mapping_1.isnull().sum())
```

```
Warehouse Pincode    0
Customer Pincode     0
Zone                 0
dtype: int64
```

```
In [ ]:
```

## Junção dos Dataframes "Relatório de Pedidos do Site" e "Lista Mestre de Produtos" baseado no "SKU"

```
In [38]: merged_order_master = pd.merge(order_report_1, sku_master_1, on='SKU')
```

```
In [39]: print(merged_order_master)
```

|     | ExternOrderNo | SKU           | Order Qty | Weight (g) |
|-----|---------------|---------------|-----------|------------|
| 0   | 2001827036    | 8904223818706 | 1.0       | 127        |
| 1   | 2001821995    | 8904223818706 | 1.0       | 127        |
| 2   | 2001819252    | 8904223818706 | 1.0       | 127        |
| 3   | 2001816996    | 8904223818706 | 1.0       | 127        |
| 4   | 2001814580    | 8904223818706 | 1.0       | 127        |
| ..  | ...           | ...           | ...       | ...        |
| 396 | 2001806616    | 8904223819123 | 1.0       | 250        |
| 397 | 2001806567    | 8904223815804 | 1.0       | 160        |
| 398 | 2001806567    | 8904223818577 | 1.0       | 150        |
| 399 | 2001806408    | 8904223819437 | 2.0       | 552        |
| 400 | 2001806232    | 8904223818645 | 6.0       | 137        |

```
[401 rows x 4 columns]
```

```
In [ ]:
```

## Renomeando a Coluna "ExternOrderNo" para "Order ID" no Dataframe "Order\_Master"

```
In [40]: merged_order_master = merged_order_master.rename(columns={'ExternOrderNo': 'Order ID'})
```

```
In [41]: print(merged_order_master.head())
```

|   | Order ID   | SKU           | Order Qty | Weight (g) |
|---|------------|---------------|-----------|------------|
| 0 | 2001827036 | 8904223818706 | 1.0       | 127        |
| 1 | 2001821995 | 8904223818706 | 1.0       | 127        |
| 2 | 2001819252 | 8904223818706 | 1.0       | 127        |
| 3 | 2001816996 | 8904223818706 | 1.0       | 127        |
| 4 | 2001814580 | 8904223818706 | 1.0       | 127        |

```
In [ ]:
```

## Junção do Conjunto de Dados de Mapeamento de CEPs dos Armazéns e Faturas das Empresas de Entrega

```
In [42]: # Eliminando as duplicatas no "Mapeamento de CEPs dos Armazéns"  
# Primeiro, extraímos os códigos PIN exclusivos do cliente do conjunto de dados de Mapeamento de CEPs dos Armazéns  
# e criamos um novo DataFrame chamado "abc_courier" para armazenar essas informações.
```

```
In [43]: print(pincod_mapping_1)
```

|     | Warehouse Pincode | Customer Pincode | Zone |
|-----|-------------------|------------------|------|
| 0   | 121003            | 507101           | d    |
| 1   | 121003            | 486886           | d    |
| 2   | 121003            | 532484           | d    |
| 3   | 121003            | 143001           | b    |
| 4   | 121003            | 515591           | d    |
| ..  | ...               | ...              | ...  |
| 119 | 121003            | 325207           | b    |
| 120 | 121003            | 303702           | b    |
| 121 | 121003            | 313301           | b    |
| 122 | 121003            | 173212           | e    |
| 123 | 121003            | 302020           | b    |

```
[124 rows x 3 columns]
```

```
In [44]: abc_courier = pincode_mapping_1.drop_duplicates(subset=['Customer Pincode'])
```

```
In [45]: print(abc_courier)
```

|     | Warehouse Pincode | Customer Pincode | Zone |
|-----|-------------------|------------------|------|
| 0   | 121003            | 507101           | d    |
| 1   | 121003            | 486886           | d    |
| 2   | 121003            | 532484           | d    |
| 3   | 121003            | 143001           | b    |
| 4   | 121003            | 515591           | d    |
| ..  | ...               | ...              | ...  |
| 114 | 121003            | 324008           | b    |
| 115 | 121003            | 302020           | b    |
| 119 | 121003            | 325207           | b    |
| 120 | 121003            | 303702           | b    |
| 121 | 121003            | 313301           | b    |

[108 rows x 3 columns]

```
In [46]: # Selecionando Colunas do Dataframe "Faturas das Empresas de Entrega"
# Em seguida, selecionamos colunas específicas ("Order ID", "Customer Pincode", "Type of Shipment")
# do conjunto de dados "Faturas das Empresas de Entrega" - "courier_invoice" e criamos um novo DataFrame chamado "courier_abc"
# para armazenar esse subconjunto de dados.
```

```
In [47]: courier_abc = courier_invoice[['Order ID', 'Customer Pincode', 'Type of Shipment']]
print(courier_abc)
```

|     | Order ID   | Customer Pincode | Type of Shipment        |
|-----|------------|------------------|-------------------------|
| 0   | 2001806232 | 507101           | Forward charges         |
| 1   | 2001806273 | 486886           | Forward charges         |
| 2   | 2001806408 | 532484           | Forward charges         |
| 3   | 2001806458 | 143001           | Forward charges         |
| 4   | 2001807012 | 515591           | Forward charges         |
| ..  | ...        | ...              | ...                     |
| 119 | 2001812941 | 325207           | Forward charges         |
| 120 | 2001809383 | 303702           | Forward and RTO charges |
| 121 | 2001820978 | 313301           | Forward charges         |
| 122 | 2001811475 | 173212           | Forward charges         |
| 123 | 2001811305 | 302020           | Forward charges         |

[124 rows x 3 columns]

```
In [48]: # Junção das Tabelas
# Em seguida, mesclamos o DataFrame 'courier_abc' com o DataFrame 'abc_courier' com base na coluna
# 'Customer Pincode'. Essa operação de mesclagem nos ajuda a associar os códigos PIN do cliente aos
# seus respectivos pedidos e tipos de remessa. O DataFrame resultante é denominado 'pincodes'.
```

```
In [49]: pincodes = courier_abc.merge(abc_courier, on='Customer Pincode')
print(pincodes)
```

|     | Order ID   | Customer Pincode | Type of Shipment        | Warehouse Pincode \ |
|-----|------------|------------------|-------------------------|---------------------|
| 0   | 2001806232 | 507101           | Forward charges         | 121003              |
| 1   | 2001806273 | 486886           | Forward charges         | 121003              |
| 2   | 2001806408 | 532484           | Forward charges         | 121003              |
| 3   | 2001806458 | 143001           | Forward charges         | 121003              |
| 4   | 2001807012 | 515591           | Forward charges         | 121003              |
| ..  | ...        | ...              | ...                     | ...                 |
| 119 | 2001811039 | 302020           | Forward charges         | 121003              |
| 120 | 2001811305 | 302020           | Forward charges         | 121003              |
| 121 | 2001812941 | 325207           | Forward charges         | 121003              |
| 122 | 2001809383 | 303702           | Forward and RTO charges | 121003              |
| 123 | 2001820978 | 313301           | Forward charges         | 121003              |

|     | Zone |
|-----|------|
| 0   | d    |
| 1   | d    |
| 2   | d    |
| 3   | b    |
| 4   | d    |
| ..  | ...  |
| 119 | b    |
| 120 | b    |
| 121 | b    |
| 122 | b    |
| 123 | b    |

[124 rows x 5 columns]

```
In [ ]:
```

## Junção dos Dataframes ('order\_master\_2' e 'pincodes')

```
In [50]: order_master_2 = merged_order_master.merge(pincodes, on='Order ID')
```

```
In [51]: print(order_master_2)
```

|     | Order ID   | SKU           | Order Qty | Weight (g) | Customer Pincode \ |
|-----|------------|---------------|-----------|------------|--------------------|
| 0   | 2001827036 | 8904223818706 | 1.0       | 127        | 173213             |
| 1   | 2001827036 | 8904223819093 | 1.0       | 150        | 173213             |
| 2   | 2001827036 | 8904223819109 | 1.0       | 100        | 173213             |
| 3   | 2001827036 | 8904223818430 | 1.0       | 165        | 173213             |
| 4   | 2001827036 | 8904223819277 | 1.0       | 350        | 173213             |
| ..  | ...        | ...           | ...       | ...        | ...                |
| 396 | 2001806251 | 8904223819161 | 1.0       | 115        | 743263             |
| 397 | 2001806251 | 8904223819260 | 1.0       | 130        | 743263             |
| 398 | 2001806233 | 8904223819161 | 1.0       | 115        | 263139             |
| 399 | 2001806233 | 8904223819260 | 1.0       | 130        | 263139             |
| 400 | 2001807328 | 8904223818997 | 1.0       | 490        | 335502             |

|     | Type of Shipment | Warehouse | Pincode | Zone |
|-----|------------------|-----------|---------|------|
| 0   | Forward charges  |           | 121003  | e    |
| 1   | Forward charges  |           | 121003  | e    |
| 2   | Forward charges  |           | 121003  | e    |
| 3   | Forward charges  |           | 121003  | e    |
| 4   | Forward charges  |           | 121003  | e    |
| ..  | ...              |           | ...     | ...  |
| 396 | Forward charges  |           | 121003  | d    |
| 397 | Forward charges  |           | 121003  | d    |
| 398 | Forward charges  |           | 121003  | b    |
| 399 | Forward charges  |           | 121003  | b    |
| 400 | Forward charges  |           | 121003  | b    |

[401 rows x 8 columns]

```
In [ ]:
```

**Calcular o Peso em Quilogramas Dividindo a Coluna 'Weight (g)' no DataFrame 'order\_master\_2' por 1000:**

```
In [52]: order_master_2['Weights (Kgs)'] = order_master_2['Weight (g)'] / 1000
```

```
In [53]: print(order_master_2)
```

|     | Order ID   | SKU           | Order Qty | Weight (g) | Customer Pincode \ |
|-----|------------|---------------|-----------|------------|--------------------|
| 0   | 2001827036 | 8904223818706 | 1.0       | 127        | 173213             |
| 1   | 2001827036 | 8904223819093 | 1.0       | 150        | 173213             |
| 2   | 2001827036 | 8904223819109 | 1.0       | 100        | 173213             |
| 3   | 2001827036 | 8904223818430 | 1.0       | 165        | 173213             |
| 4   | 2001827036 | 8904223819277 | 1.0       | 350        | 173213             |
| ..  | ...        | ...           | ...       | ...        | ...                |
| 396 | 2001806251 | 8904223819161 | 1.0       | 115        | 743263             |
| 397 | 2001806251 | 8904223819260 | 1.0       | 130        | 743263             |
| 398 | 2001806233 | 8904223819161 | 1.0       | 115        | 263139             |
| 399 | 2001806233 | 8904223819260 | 1.0       | 130        | 263139             |
| 400 | 2001807328 | 8904223818997 | 1.0       | 490        | 335502             |

|     | Type of Shipment | Warehouse | Pincode | Zone | Weights (Kgs) |
|-----|------------------|-----------|---------|------|---------------|
| 0   | Forward charges  |           | 121003  | e    | 0.127         |
| 1   | Forward charges  |           | 121003  | e    | 0.150         |
| 2   | Forward charges  |           | 121003  | e    | 0.100         |
| 3   | Forward charges  |           | 121003  | e    | 0.165         |
| 4   | Forward charges  |           | 121003  | e    | 0.350         |
| ..  | ...              |           | ...     | ...  | ...           |
| 396 | Forward charges  |           | 121003  | d    | 0.115         |
| 397 | Forward charges  |           | 121003  | d    | 0.130         |
| 398 | Forward charges  |           | 121003  | b    | 0.115         |
| 399 | Forward charges  |           | 121003  | b    | 0.130         |
| 400 | Forward charges  |           | 121003  | b    | 0.490         |

[401 rows x 9 columns]

In [ ]:

## Calcular o Peso da Remessa Usando a Regra de Arredondamento

In [54]: *# A função weight\_slab() é definida para determinar a faixa de peso com base no peso da remessa.  
# Ela pega o peso de entrada e aplica certas condições para calcular a faixa de peso.*

In [55]: *# A função abaixo funciona da seguinte maneira:  
##  
## 1. A função primeiro calcula o restante do peso dividido por 1 e arredonda para uma casa decimal.  
## Se o restante for 0,0, significa que o peso é um múltiplo de 1 KG, e a função retorna o peso como ele é;  
##  
## 2. Se o restante for maior que 0,5, significa que o peso excede a próxima a faixa de meio KG.  
## Nesse caso, a função arredonda o peso para o inteiro mais próximo e adiciona 1,0 a ele, o que representa a*

```
## próxima "faixa" mais pesada;
##
## 3. Se o restante for menor ou igual a 0,5, significa que o peso cai na faixa atual de meio KG.
## Nesse caso, a função arredonda o peso para o inteiro mais próximo e adiciona 0,5 a ele, o que representa a
## placa de peso atual.
```

```
In [56]: def weight_slab(weight):
        i = round(weight % 1, 1)
        if i == 0.0:
            return weight
        elif i > 0.5:
            return int(weight) + 1.0
        else:
            return int(weight) + 0.5
```

```
In [57]: # Aplicando a função de arredondamento, criando uma nova coluna no dataframe principal
```

```
In [58]: order_master_2['Weight Slab (KG)'] = order_master_2['Weights (Kgs)'].apply(weight_slab)
```

```
In [59]: print(order_master_2)
```



|     | Order ID   | SKU           | Order Qty | Weight (g) | Customer Pincode \ |
|-----|------------|---------------|-----------|------------|--------------------|
| 0   | 2001827036 | 8904223818706 | 1.0       | 127        | 173213             |
| 1   | 2001827036 | 8904223819093 | 1.0       | 150        | 173213             |
| 2   | 2001827036 | 8904223819109 | 1.0       | 100        | 173213             |
| 3   | 2001827036 | 8904223818430 | 1.0       | 165        | 173213             |
| 4   | 2001827036 | 8904223819277 | 1.0       | 350        | 173213             |
| ..  | ...        | ...           | ...       | ...        | ...                |
| 396 | 2001806251 | 8904223819161 | 1.0       | 115        | 743263             |
| 397 | 2001806251 | 8904223819260 | 1.0       | 130        | 743263             |
| 398 | 2001806233 | 8904223819161 | 1.0       | 115        | 263139             |
| 399 | 2001806233 | 8904223819260 | 1.0       | 130        | 263139             |
| 400 | 2001807328 | 8904223818997 | 1.0       | 490        | 335502             |

|     | Type of Shipment | Warehouse | Pincode | Zone | Weights (Kgs) | Weight Slab (KG) |
|-----|------------------|-----------|---------|------|---------------|------------------|
| 0   | Forward charges  |           | 121003  | e    | 0.127         | 0.5              |
| 1   | Forward charges  |           | 121003  | e    | 0.150         | 0.5              |
| 2   | Forward charges  |           | 121003  | e    | 0.100         | 0.5              |
| 3   | Forward charges  |           | 121003  | e    | 0.165         | 0.5              |
| 4   | Forward charges  |           | 121003  | e    | 0.350         | 0.5              |
| ..  | ...              |           | ...     | ...  | ...           | ...              |
| 396 | Forward charges  |           | 121003  | d    | 0.115         | 0.5              |
| 397 | Forward charges  |           | 121003  | d    | 0.130         | 0.5              |
| 398 | Forward charges  |           | 121003  | b    | 0.115         | 0.5              |
| 399 | Forward charges  |           | 121003  | b    | 0.130         | 0.5              |
| 400 | Forward charges  |           | 121003  | b    | 0.490         | 0.5              |

[401 rows x 10 columns]

```
In [60]: ## Aplicando a função de arredondamento, criando uma nova coluna no dataframe "Faturas das Empresas de Entrega"-courier_invoice
```

```
In [61]: courier_invoice['Weight Slab Charged by Courier Company']=(courier_invoice['Charged Weight']).apply(weight_slab)
```

```
In [62]: print(courier_invoice)
```

|     | AWB Code      | Order ID   | Charged Weight | Warehouse Pincode \ |
|-----|---------------|------------|----------------|---------------------|
| 0   | 1091117222124 | 2001806232 | 1.30           | 121003              |
| 1   | 1091117222194 | 2001806273 | 1.00           | 121003              |
| 2   | 1091117222931 | 2001806408 | 2.50           | 121003              |
| 3   | 1091117223244 | 2001806458 | 1.00           | 121003              |
| 4   | 1091117229345 | 2001807012 | 0.15           | 121003              |
| ..  | ...           | ...        | ...            | ...                 |
| 119 | 1091118551656 | 2001812941 | 0.73           | 121003              |
| 120 | 1091117614452 | 2001809383 | 0.50           | 121003              |
| 121 | 1091120922803 | 2001820978 | 0.50           | 121003              |
| 122 | 1091121844806 | 2001811475 | 0.50           | 121003              |
| 123 | 1091121846136 | 2001811305 | 0.50           | 121003              |

|     | Customer Pincode | Zone | Type of Shipment        | Billing Amount (Rs.) \ |
|-----|------------------|------|-------------------------|------------------------|
| 0   | 507101           | d    | Forward charges         | 135.0                  |
| 1   | 486886           | d    | Forward charges         | 90.2                   |
| 2   | 532484           | d    | Forward charges         | 224.6                  |
| 3   | 143001           | b    | Forward charges         | 61.3                   |
| 4   | 515591           | d    | Forward charges         | 45.4                   |
| ..  | ...              | ...  | ...                     | ...                    |
| 119 | 325207           | d    | Forward charges         | 90.2                   |
| 120 | 303702           | d    | Forward and RTO charges | 86.7                   |
| 121 | 313301           | d    | Forward charges         | 45.4                   |
| 122 | 173212           | b    | Forward charges         | 33.0                   |
| 123 | 302020           | d    | Forward charges         | 45.4                   |

|     | Weight Slab Charged by Courier Company |
|-----|----------------------------------------|
| 0   | 1.5                                    |
| 1   | 1.0                                    |
| 2   | 2.5                                    |
| 3   | 1.0                                    |
| 4   | 0.5                                    |
| ..  | ...                                    |
| 119 | 1.0                                    |
| 120 | 0.5                                    |
| 121 | 0.5                                    |
| 122 | 0.5                                    |
| 123 | 0.5                                    |

[124 rows x 9 columns]

In [ ]:

## Renomeando Colunas dos Dataframes

```
In [63]: # Dataframe "Faturas das Empresas de Entrega"- courier_invoice
```

```
In [64]: courier_invoice = courier_invoice.rename(columns={'Zone': 'Delivery Zone Charged by Courier Company'})
```

```
In [65]: print(courier_invoice)
```

|     | AWB Code      | Order ID   | Charged Weight | Warehouse Pincode \ |
|-----|---------------|------------|----------------|---------------------|
| 0   | 1091117222124 | 2001806232 | 1.30           | 121003              |
| 1   | 1091117222194 | 2001806273 | 1.00           | 121003              |
| 2   | 1091117222931 | 2001806408 | 2.50           | 121003              |
| 3   | 1091117223244 | 2001806458 | 1.00           | 121003              |
| 4   | 1091117229345 | 2001807012 | 0.15           | 121003              |
| ..  | ...           | ...        | ...            | ...                 |
| 119 | 1091118551656 | 2001812941 | 0.73           | 121003              |
| 120 | 1091117614452 | 2001809383 | 0.50           | 121003              |
| 121 | 1091120922803 | 2001820978 | 0.50           | 121003              |
| 122 | 1091121844806 | 2001811475 | 0.50           | 121003              |
| 123 | 1091121846136 | 2001811305 | 0.50           | 121003              |

|     | Customer Pincode | Delivery Zone | Charged by Courier Company \ |
|-----|------------------|---------------|------------------------------|
| 0   | 507101           |               | d                            |
| 1   | 486886           |               | d                            |
| 2   | 532484           |               | d                            |
| 3   | 143001           |               | b                            |
| 4   | 515591           |               | d                            |
| ..  | ...              |               | ...                          |
| 119 | 325207           |               | d                            |
| 120 | 303702           |               | d                            |
| 121 | 313301           |               | d                            |
| 122 | 173212           |               | b                            |
| 123 | 302020           |               | d                            |

|     | Type of Shipment        | Billing Amount (Rs.) \ |
|-----|-------------------------|------------------------|
| 0   | Forward charges         | 135.0                  |
| 1   | Forward charges         | 90.2                   |
| 2   | Forward charges         | 224.6                  |
| 3   | Forward charges         | 61.3                   |
| 4   | Forward charges         | 45.4                   |
| ..  | ...                     | ...                    |
| 119 | Forward charges         | 90.2                   |
| 120 | Forward and RTO charges | 86.7                   |
| 121 | Forward charges         | 45.4                   |
| 122 | Forward charges         | 33.0                   |
| 123 | Forward charges         | 45.4                   |

|   | Weight Slab Charged by Courier Company |
|---|----------------------------------------|
| 0 | 1.5                                    |
| 1 | 1.0                                    |
| 2 | 2.5                                    |
| 3 | 1.0                                    |

|     |     |
|-----|-----|
| 4   | 0.5 |
| ..  | ... |
| 119 | 1.0 |
| 120 | 0.5 |
| 121 | 0.5 |
| 122 | 0.5 |
| 123 | 0.5 |

[124 rows x 9 columns]

```
In [66]: # Dataframe Principal - order_master_2
```

```
In [67]: order_master_2 = order_master_2.rename(columns={'Zone': 'Delivery Zone As Per ABC'})
order_master_2 = order_master_2.rename(columns={'Weight Slab (KG)': 'Weight Slab As Per ABC'})
```

```
In [68]: print(order_master_2)
```

|     | Order ID   | SKU           | Order Qty | Weight (g) | Customer Pincode \ |
|-----|------------|---------------|-----------|------------|--------------------|
| 0   | 2001827036 | 8904223818706 | 1.0       | 127        | 173213             |
| 1   | 2001827036 | 8904223819093 | 1.0       | 150        | 173213             |
| 2   | 2001827036 | 8904223819109 | 1.0       | 100        | 173213             |
| 3   | 2001827036 | 8904223818430 | 1.0       | 165        | 173213             |
| 4   | 2001827036 | 8904223819277 | 1.0       | 350        | 173213             |
| ..  | ...        | ...           | ...       | ...        | ...                |
| 396 | 2001806251 | 8904223819161 | 1.0       | 115        | 743263             |
| 397 | 2001806251 | 8904223819260 | 1.0       | 130        | 743263             |
| 398 | 2001806233 | 8904223819161 | 1.0       | 115        | 263139             |
| 399 | 2001806233 | 8904223819260 | 1.0       | 130        | 263139             |
| 400 | 2001807328 | 8904223818997 | 1.0       | 490        | 335502             |

|     | Type of Shipment | Warehouse Pincode | Delivery Zone | As Per ABC \ |
|-----|------------------|-------------------|---------------|--------------|
| 0   | Forward charges  | 121003            |               | e            |
| 1   | Forward charges  | 121003            |               | e            |
| 2   | Forward charges  | 121003            |               | e            |
| 3   | Forward charges  | 121003            |               | e            |
| 4   | Forward charges  | 121003            |               | e            |
| ..  | ...              | ...               |               | ...          |
| 396 | Forward charges  | 121003            |               | d            |
| 397 | Forward charges  | 121003            |               | d            |
| 398 | Forward charges  | 121003            |               | b            |
| 399 | Forward charges  | 121003            |               | b            |
| 400 | Forward charges  | 121003            |               | b            |

|     | Weights (Kgs) | Weight Slab | As Per ABC |
|-----|---------------|-------------|------------|
| 0   | 0.127         |             | 0.5        |
| 1   | 0.150         |             | 0.5        |
| 2   | 0.100         |             | 0.5        |
| 3   | 0.165         |             | 0.5        |
| 4   | 0.350         |             | 0.5        |
| ..  | ...           |             | ...        |
| 396 | 0.115         |             | 0.5        |
| 397 | 0.130         |             | 0.5        |
| 398 | 0.115         |             | 0.5        |
| 399 | 0.130         |             | 0.5        |
| 400 | 0.490         |             | 0.5        |

[401 rows x 10 columns]

In [ ]:

## Calcular o Valor Esperado para as Remessas

```
In [69]: # Criando uma lista vazia para armazenar os resultados
```

```
In [70]: total_expected_charge = []
```

```
In [71]: # Neste código, fazemos um loop em cada linha do DataFrame "order_master_2" para calcular as cobranças  
# esperadas com base nas tarifas da ABC.
```

```
In [72]: # O código abaixo funciona da seguinte maneira:  
## 1. Recuperamos as taxas e parâmetros necessários, como cobranças fixas e sobretaxas por nível de peso  
##    para remessas forward e RTO, com base na área de entrega;  
## 2. Em seguida, determinamos a faixa de peso para cada linha. Se o tipo de remessa for 'Forward Charges',  
##    calculamos o peso adicional além da placa de peso básico (0,5 KG) e aplicamos as cobranças adicionais correspondentes.  
##    Para remessas "Forward Charges and RTO Charges", consideramos cobranças adicionais para componentes de prazo e RTO.  
## 3. Por fim, armazenamos as cobranças esperadas calculadas na coluna "Expected charges according to ABC" do DataFrame  
##    "order_master_2". Isso nos permite comparar as cobranças esperadas com as cobranças faturadas para analisar a  
##    precisão das cobranças da empresa de entrega.
```

```
In [73]: for _, row in order_master_2.iterrows():  
    fwd_category = 'fwd_' + row['Delivery Zone As Per ABC']  
    fwd_fixed = courier_company_rates.at[0, fwd_category + '_fixed']  
    fwd_additional = courier_company_rates.at[0, fwd_category + '_additional']  
    rto_category = 'rto_' + row['Delivery Zone As Per ABC']  
    rto_fixed = courier_company_rates.at[0, rto_category + '_fixed']  
    rto_additional = courier_company_rates.at[0, rto_category + '_additional']  
  
    weight_slab = row['Weight Slab As Per ABC']  
  
    if row['Type of Shipment'] == 'Forward charges':  
        additional_weight = max(0, (weight_slab - 0.5) / 0.5)  
        total_expected_charge.append(fwd_fixed + additional_weight * fwd_additional)  
    elif row['Type of Shipment'] == 'Forward and RTO charges':  
        additional_weight = max(0, (weight_slab - 0.5) / 0.5)  
        total_expected_charge.append(fwd_fixed + additional_weight * (fwd_additional + rto_additional))  
    else:  
        total_expected_charge.append(0)
```

```
In [74]: # Criando a coluna no Dataframe Principal com as cobranças esperadas com base nas tarifas da ABC.
```

```
In [75]: order_master_2['Expected Charge as per ABC'] = total_expected_charge
```

```
In [76]: print(order_master_2.head())
```

|   | Order ID   | SKU           | Order Qty | Weight (g) | Customer Pincode | \ |
|---|------------|---------------|-----------|------------|------------------|---|
| 0 | 2001827036 | 8904223818706 | 1.0       | 127        | 173213           |   |
| 1 | 2001827036 | 8904223819093 | 1.0       | 150        | 173213           |   |
| 2 | 2001827036 | 8904223819109 | 1.0       | 100        | 173213           |   |
| 3 | 2001827036 | 8904223818430 | 1.0       | 165        | 173213           |   |
| 4 | 2001827036 | 8904223819277 | 1.0       | 350        | 173213           |   |

|   | Type of Shipment | Warehouse Pincode | Delivery Zone | As Per ABC | Weights (Kgs) | \ |
|---|------------------|-------------------|---------------|------------|---------------|---|
| 0 | Forward charges  | 121003            |               | e          | 0.127         |   |
| 1 | Forward charges  | 121003            |               | e          | 0.150         |   |
| 2 | Forward charges  | 121003            |               | e          | 0.100         |   |
| 3 | Forward charges  | 121003            |               | e          | 0.165         |   |
| 4 | Forward charges  | 121003            |               | e          | 0.350         |   |

|   | Weight Slab | As Per ABC | Expected Charge as per ABC |
|---|-------------|------------|----------------------------|
| 0 |             | 0.5        | 56.6                       |
| 1 |             | 0.5        | 56.6                       |
| 2 |             | 0.5        | 56.6                       |
| 3 |             | 0.5        | 56.6                       |
| 4 |             | 0.5        | 56.6                       |

```
In [ ]:
```

## Junção dos Dataframes - Principal "order\_master\_2" e Faturas das Empresas de Entrega "courier\_invoice"

```
In [77]: order_master_final = order_master_2.merge(courier_invoice, on='Order ID')
```

```
In [78]: print(order_master_final.head())
```



|   | Order ID   | SKU           | Order Qty | Weight (g) | Customer Pincode_x \ |
|---|------------|---------------|-----------|------------|----------------------|
| 0 | 2001827036 | 8904223818706 | 1.0       | 127        | 173213               |
| 1 | 2001827036 | 8904223819093 | 1.0       | 150        | 173213               |
| 2 | 2001827036 | 8904223819109 | 1.0       | 100        | 173213               |
| 3 | 2001827036 | 8904223818430 | 1.0       | 165        | 173213               |
| 4 | 2001827036 | 8904223819277 | 1.0       | 350        | 173213               |

|   | Type of Shipment_x | Warehouse Pincode_x | Delivery Zone As Per ABC \ |
|---|--------------------|---------------------|----------------------------|
| 0 | Forward charges    | 121003              | e                          |
| 1 | Forward charges    | 121003              | e                          |
| 2 | Forward charges    | 121003              | e                          |
| 3 | Forward charges    | 121003              | e                          |
| 4 | Forward charges    | 121003              | e                          |

|   | Weights (Kgs) | Weight Slab As Per ABC | Expected Charge as per ABC \ |
|---|---------------|------------------------|------------------------------|
| 0 | 0.127         | 0.5                    | 56.6                         |
| 1 | 0.150         | 0.5                    | 56.6                         |
| 2 | 0.100         | 0.5                    | 56.6                         |
| 3 | 0.165         | 0.5                    | 56.6                         |
| 4 | 0.350         | 0.5                    | 56.6                         |

|   | AWB Code      | Charged Weight | Warehouse Pincode_y | Customer Pincode_y \ |
|---|---------------|----------------|---------------------|----------------------|
| 0 | 1091122418320 | 1.6            | 121003              | 173213               |
| 1 | 1091122418320 | 1.6            | 121003              | 173213               |
| 2 | 1091122418320 | 1.6            | 121003              | 173213               |
| 3 | 1091122418320 | 1.6            | 121003              | 173213               |
| 4 | 1091122418320 | 1.6            | 121003              | 173213               |

|   | Delivery Zone Charged by Courier Company | Type of Shipment_y \ |
|---|------------------------------------------|----------------------|
| 0 | b                                        | Forward charges      |
| 1 | b                                        | Forward charges      |
| 2 | b                                        | Forward charges      |
| 3 | b                                        | Forward charges      |
| 4 | b                                        | Forward charges      |

|   | Billing Amount (Rs.) | Weight Slab Charged by Courier Company |
|---|----------------------|----------------------------------------|
| 0 | 117.9                | 2.0                                    |
| 1 | 117.9                | 2.0                                    |
| 2 | 117.9                | 2.0                                    |
| 3 | 117.9                | 2.0                                    |
| 4 | 117.9                | 2.0                                    |

In [ ]:

## Criando a Tabela da Solução Final Conforme Solicitado

```
In [79]: # Criando a Dataframe Final
```

```
In [80]: df_table_solution = order_master_final[['Order ID',  
                                                'Weights (Kgs)',  
                                                'Weight Slab As Per ABC',  
                                                'Delivery Zone As Per ABC',  
                                                'Expected Charge as per ABC',  
                                                'AWB Code',  
                                                'Charged Weight',  
                                                'Delivery Zone Charged by Courier Company',  
                                                'Billing Amount (Rs.)',  
                                                'Weight Slab Charged by Courier Company'  
                                                ]]
```

```
In [81]: df_table_solution = df_table_solution.rename(columns={'Weights (Kgs)': 'Total Weight (Kg)'})  
df_table_solution = df_table_solution.rename(columns={'Weight Slab As Per ABC': 'Weight Slab (Kg)'})  
df_table_solution = df_table_solution.rename(columns={'Delivery Zone As Per ABC': 'Delivery Zone as per ABC'})  
df_table_solution = df_table_solution.rename(columns={'Expected Charge as per ABC': 'Expected Charge as per ABC (Rs.)'})  
df_table_solution = df_table_solution.rename(columns={'AWB Code': 'AWB Number'})  
df_table_solution = df_table_solution.rename(columns={'Charged Weight': 'Total Weight as per Courier Company (Kg)'})  
df_table_solution = df_table_solution.rename(columns={'Weight Slab Charged by Courier Company': 'Weight Slab Charged by Courier C
```

```
In [82]: display(df_table_solution)
```

|     | Order ID   | Total Weight (Kg) | Weight Slab (Kg) | Delivery Zone as per ABC | Expected Charge as per ABC (Rs.) | AWB Number    | Total Weight as per Courier Company (Kg) | Delivery Zone Charged by Courier Company | Billing Amount (Rs.) | Weight Slab Charged by Courier Company (Kg) |
|-----|------------|-------------------|------------------|--------------------------|----------------------------------|---------------|------------------------------------------|------------------------------------------|----------------------|---------------------------------------------|
| 0   | 2001827036 | 0.127             | 0.5              | e                        | 56.6                             | 1091122418320 | 1.60                                     | b                                        | 117.9                | 2.0                                         |
| 1   | 2001827036 | 0.150             | 0.5              | e                        | 56.6                             | 1091122418320 | 1.60                                     | b                                        | 117.9                | 2.0                                         |
| 2   | 2001827036 | 0.100             | 0.5              | e                        | 56.6                             | 1091122418320 | 1.60                                     | b                                        | 117.9                | 2.0                                         |
| 3   | 2001827036 | 0.165             | 0.5              | e                        | 56.6                             | 1091122418320 | 1.60                                     | b                                        | 117.9                | 2.0                                         |
| 4   | 2001827036 | 0.350             | 0.5              | e                        | 56.6                             | 1091122418320 | 1.60                                     | b                                        | 117.9                | 2.0                                         |
| ... | ...        | ...               | ...              | ...                      | ...                              | ...           | ...                                      | ...                                      | ...                  | ...                                         |
| 396 | 2001806251 | 0.115             | 0.5              | d                        | 45.4                             | 1091117222146 | 1.27                                     | d                                        | 135.0                | 1.5                                         |
| 397 | 2001806251 | 0.130             | 0.5              | d                        | 45.4                             | 1091117222146 | 1.27                                     | d                                        | 135.0                | 1.5                                         |
| 398 | 2001806233 | 0.115             | 0.5              | b                        | 33.0                             | 1091117222135 | 0.78                                     | b                                        | 61.3                 | 1.0                                         |
| 399 | 2001806233 | 0.130             | 0.5              | b                        | 33.0                             | 1091117222135 | 0.78                                     | b                                        | 61.3                 | 1.0                                         |
| 400 | 2001807328 | 0.490             | 0.5              | b                        | 33.0                             | 1091117324346 | 2.28                                     | d                                        | 224.6                | 2.5                                         |

401 rows × 10 columns

In [ ]:

## Calculando a Diferença entre os Preços Cobrados e os Preços Esperados

In [83]: `df_diff = order_master_final`

In [84]: `df_diff['Difference (Rs.)'] = df_diff['Billing Amount (Rs.)'] - df_diff['Expected Charge as per ABC']`

In [85]: `df_final = df_diff[['Order ID', 'Expected Charge as per ABC', 'Billing Amount (Rs.)', 'Difference (Rs.)']]`

In [86]: `print(df_final)`

|     | Order ID   | Expected Charge as per ABC | Billing Amount (Rs.) \ |
|-----|------------|----------------------------|------------------------|
| 0   | 2001827036 | 56.6                       | 117.9                  |
| 1   | 2001827036 | 56.6                       | 117.9                  |
| 2   | 2001827036 | 56.6                       | 117.9                  |
| 3   | 2001827036 | 56.6                       | 117.9                  |
| 4   | 2001827036 | 56.6                       | 117.9                  |
| ..  | ...        | ...                        | ...                    |
| 396 | 2001806251 | 45.4                       | 135.0                  |
| 397 | 2001806251 | 45.4                       | 135.0                  |
| 398 | 2001806233 | 33.0                       | 61.3                   |
| 399 | 2001806233 | 33.0                       | 61.3                   |
| 400 | 2001807328 | 33.0                       | 224.6                  |

|     | Difference (Rs.) |
|-----|------------------|
| 0   | 61.3             |
| 1   | 61.3             |
| 2   | 61.3             |
| 3   | 61.3             |
| 4   | 61.3             |
| ..  | ...              |
| 396 | 89.6             |
| 397 | 89.6             |
| 398 | 28.3             |
| 399 | 28.3             |
| 400 | 191.6            |

[401 rows x 4 columns]

In [ ]:

## Calculando a Precisão das Taxas de Entrega B2B com Base nos Preços Cobrados e nos Preços Esperados

In [87]: *# Calculando a Quantidade de Taxas em Cada Categoria*

In [88]: `total_correctly_charged = len(df_final[df_final['Difference (Rs.)'] == 0])`

In [89]: `total_overcharged = len(df_final[df_final['Difference (Rs.)'] > 0])`

In [90]: `total_undercharged = len(df_final[df_final['Difference (Rs.)'] < 0])`

```
In [91]: # Calculando o Total Acumulado (Rs.) em Cada Categoria
```

```
In [92]: amount_correctly_charged = df_final[df_final['Difference (Rs.)'] == 0]['Expected Charge as per ABC'].sum()
```

```
In [93]: amount_overcharged = abs(df_final[df_final['Difference (Rs.)'] > 0]['Difference (Rs.)'].sum())
```

```
In [94]: amount_undercharged = df_final[df_final['Difference (Rs.)'] < 0]['Difference (Rs.)'].sum()
```

```
In [ ]:
```

## Criando a Tabela para Realizar a Análise da Precisão das Taxas Cobradas pela Empresa de Entrega

```
In [95]: # Criando um Novo DataFrame para Sumarização
```

```
In [96]: summary_data = {'Description': ['Total Orders where ABC has been correctly charged',  
                                         'Total Orders where ABC has been overcharged',  
                                         'Total Orders where ABC has been undercharged'],  
                        'Count': [total_correctly_charged, total_overcharged, total_undercharged],  
                        'Amount (Rs.)': [amount_correctly_charged, amount_overcharged, amount_undercharged]}
```

```
In [97]: df_summary = pd.DataFrame(summary_data)
```

```
In [98]: display(df_summary)
```

|   | Description                                       | Count | Amount (Rs.) |
|---|---------------------------------------------------|-------|--------------|
| 0 | Total Orders where ABC has been correctly charged | 12    | 507.6        |
| 1 | Total Orders where ABC has been overcharged       | 382   | 33750.5      |
| 2 | Total Orders where ABC has been undercharged      | 7     | -165.2       |

```
In [ ]:
```

## Criando um Gráfico para Verificar a Proporção de Erros

```
In [99]: fig = px.pie(df_summary,
                    values='Count',
                    names='Description',
                    color='Description',
                    color_discrete_map={'Total Orders where ABC has been correctly charged':'cyan',
                                         'Total Orders where ABC has been overcharged':'royalblue',
                                         'Total Orders where ABC has been undercharged':'purple'},

                    hole=0.4,
                    title='Orders Error Proportion')
```

```
In [100... # Customizações do Gráfico
```

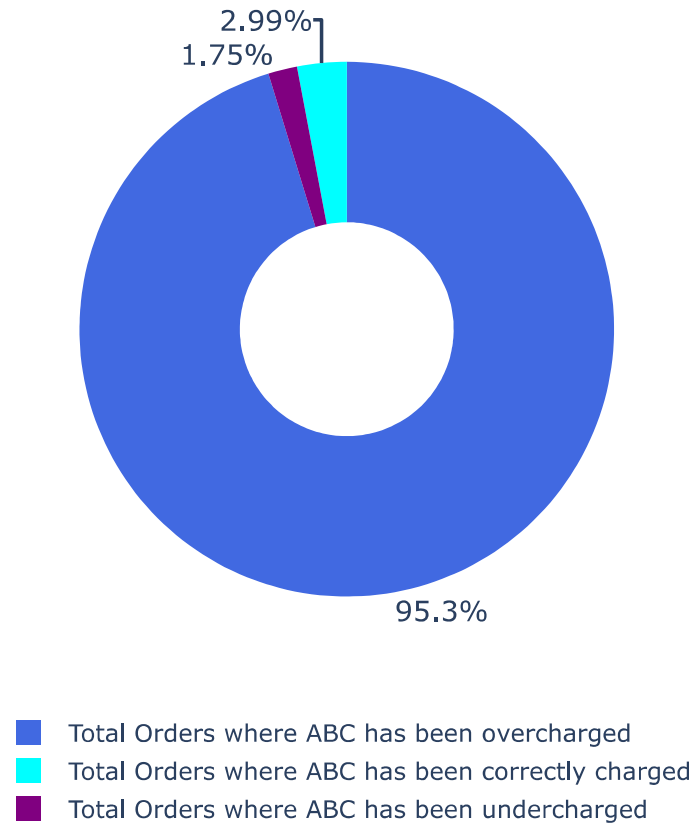
```
In [101... # Customizando o título
fig.update_layout(title={'text': 'Orders Error Proportion',
                          'font': {'size': 28, 'color': 'blue'}})

# Posicionando a Legenda abaixo do gráfico
fig.update_layout(legend=dict(
    orientation="h",      # Orientação horizontal
    yanchor="top",        # Ancoragem superior
    y=-0.2,               # Posiciona a Legenda abaixo do gráfico
    xanchor="center",     # Ancoragem no centro
    x= 0.5                # Centraliza a Legenda horizontalmente
))

# Ajustando os Labels das porcentagens
fig.update_traces(textposition='outside',      # Coloca os Labels fora do gráfico
                  textinfo='percent',          # Mostra a porcentagem
                  textfont_size=14)            # Aumenta o tamanho da fonte dos Labels

#Ajustando o tamanho do gráfico
fig.update_layout(width=500, height=500)      # Define a Largura e a altura do gráfico
```

# Orders Error Proportion



In [ ]:

## Conclusão

Neste projeto, exploramos como avaliar a precisão das taxas de entrega em transações B2B utilizando Python. O foco da análise foi garantir que as empresas estejam sendo cobradas de forma justa e precisa pelos serviços de entrega contratados. A análise de precisão das taxas de entrega é essencial para a transparência e eficiência nas operações comerciais, assegurando que os custos estejam alinhados com os serviços efetivamente prestados.

Espero que este projeto didático tenha sido útil para você.

Referência do Projeto: <https://thecleverprogrammer.com/2023/05/22/b2b-courier-charges-accuracy-analysis-using-python/>

Descrição do Problema e Base de Dados: <https://statso.io/b2b-ecommerce-fraud-case-study/>

Autor: Aman Kharwal - <https://amankharwal.medium.com/>

Projeto Estudado por: Giovani Aloísio da Luz

Disponível em: <https://www.giovani-luz.com.br/>

In [ ]: