

## ▼ **Análisis, Visualización y Curación de Datos**

### Consigna 2

Determinar la cantidad de diálogos del dataset, cantidad de tutores, y cantidad de estudiantes. Determinar cuántos turnos hay del tutor y cuántos del estudiante en total y en promedio por diálogo. Graficar la distribución.

### Importación de módulos


---

```
1 import os
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import seaborn as sns
6
7 from columns_to_keep import C2K
8 from type_to_fix import T2F
```

## ▼ **Carga de la tabla de metadata**


---

```
1 data_dir = os.path.join('.', 'dataset')
2 data_file_name = 'datadump-20150801-20171219.csv'
3 full_data_file_name = os.path.join(data_dir, data_file_name)
4 df = pd.read_csv(full_data_file_name)
5 display(df.shape)
```

 (63265, 111)

## ▼ **Preprocesamiento o curación de la metadata**

```
1 # Remuevo espacios al inicio y final del nombre de la columna
2 df.columns = [column.strip() for column in df.columns]
3
4 # Tomo solo las columnas que vamos a necesitar
5 df = df[C2K]
6
7 # Se corrigen los tipos de los datos
8 df = df.astype(T2F)
9
10 display(df.shape)
```


 (63265, 39)

## ▼ **Carga la tabla de mensajes preprocesados**

---

### Tamaño del dataset de train

```
1 data_file_name = 'train_yup_messages_preprocessed.csv'
2 full_data_file_name = os.path.join(data_dir, data_file_name)
3 df_messages = pd.read_csv(full_data_file_name)
4 print("El dataset de train con los mensajes preprocesados tiene", df_messages.shape[0], " filas y "
```

 (7441, 6)

```
1 print("El dataset de train con los mensajes preprocesados tiene", df_messages.shape[0], " filas y "
```

➞ El dataset de train con los mensajes preprocesados tiene 1102917 filas y 6 columnas

## ▼ Cálculos

A continuación, se calcula la cantidad de sesiones únicas del dataset de mensajes preprocesados y la cantidad de sesiones en el dataset de metadata, los mismos no coinciden ya que actualmente en el caso de los mensajes preprocesados se trabaja con el dataset de entrenamiento en lugar de trabajar con el dataset completo. Por lo que se opta por tomar como fuente el dataset de metadata para este propósito.

```
1 n_dialogos = df_messages.session_id.unique().shape[0]
2 n_sessions = df.session_id.shape[0]
3 n_tutors = df.tutor_id.unique().shape[0]
4 n_students = df.student_id.unique().shape[0]
5
6 display(f'>> El número de diálogos computados a partir del dataset de mensajes preprocesados es de
7 display(f'>> Además la cantidad total de tutores es de {n_tutors} y la de estudiantes de {n_student
```

➞ '>> El número de diálogos computados a partir del dataset de mensajes preprocesados es de 17987, m  
el archivo de metadata es: 63265.'  
'>> Además la cantidad total de tutores es de 150 y la de estudiantes de .16808'

## ▼ Cálculo del número de turnos de estudiantes y tutores por diálogo

Pasos:

1. Se calcula el número de turnos agrupados por `session_id` y `sent_from`
2. Se descartan las filas que no son turnos del tutor ni del estudiante, es decir, que corresponden al bot.
3. Se crea una nueva columna con el cálculo realizado.
4. Se remueven las columnas no necesarias como el tipo de contenido y el texto del diálogo, etc.

```
1 st_turnos_total= df_messages.groupby(['session_id']).count()
2 print('El total de turnos es de ', st_turnos_total.shape[0], '\n')
3
4 # 1.
5 st_turnos = df_messages.groupby(['session_id','sent_from']).count().reset_index()
6
7 # 2. Turnos solamente de tutores y estudiantes
8 st_turnos = st_turnos[st_turnos.sent_from.isin(['student', 'tutor'])]
9
10 # 3.
11 st_turnos['counts'] = st_turnos['text']
12
13 # 4.
14 st_turnos.drop(columns=['created_at', 'sent_to', 'content_type', 'text'], inplace=True)
15 display(st_turnos)
```

➞

El total de turnos es de 17987 .

▼ Cálculo del número total de turnos por diálogo

Pasos:

5. Agrupar 'st\_turnos' por session\_id y luego se suman las coincidencias.

```
1 # 5.
2 turnos = st_turnos.groupby('session_id').sum()
3 display(turnos)
```

↗

| counts     |     |
|------------|-----|
| session_id |     |
| 299889     | 20  |
| 299890     | 84  |
| 299891     | 9   |
| 299892     | 49  |
| 299893     | 36  |
| ...        | ... |
| 326161     | 14  |
| 326163     | 42  |
| 326165     | 25  |
| 326166     | 14  |
| 326167     | 1   |

17938 rows × 1 columns

▼ Cálculo del promedio de turnos

- 1. Se calcula el promedio de turnos discriminado por tutor y estudiante.
- 2. Por último, se calcula el promedio total de turnos.

```
1 # 1.
2 mean_disc = st_turnos.groupby('sent_from').mean().drop(columns='session_id')
3 mean_tutors = mean_disc.loc['tutor','counts']
4 mean_students = mean_disc.loc['student','counts']
5 # se calcula la mediana sólo para conocer un poco más de la distribución
6 mediana_disc = st_turnos.groupby('sent_from').median().drop(columns='session_id')
7 print('Mediana de turnos','\n')
8 print(mediana_disc)
9
10 # 2. promedio total
11 # mean_total = turnos.counts.mean()
12 mean_total = mean_tutors + mean_students
13
14 print(f'\nEl Promedio de turnos por tutores de {mean_tutors: .2f}, mientras que el Promedio de turnos por estudiantes es de {mean_students: .2f}')
15 display(f'>> Promedio total de turnos entre estudiantes y tutores es de {mean_total:.2f}')
16
```

↗

| counts    |      |
|-----------|------|
| sent_from |      |
| student   | 14.0 |
| tutor     | 21.0 |

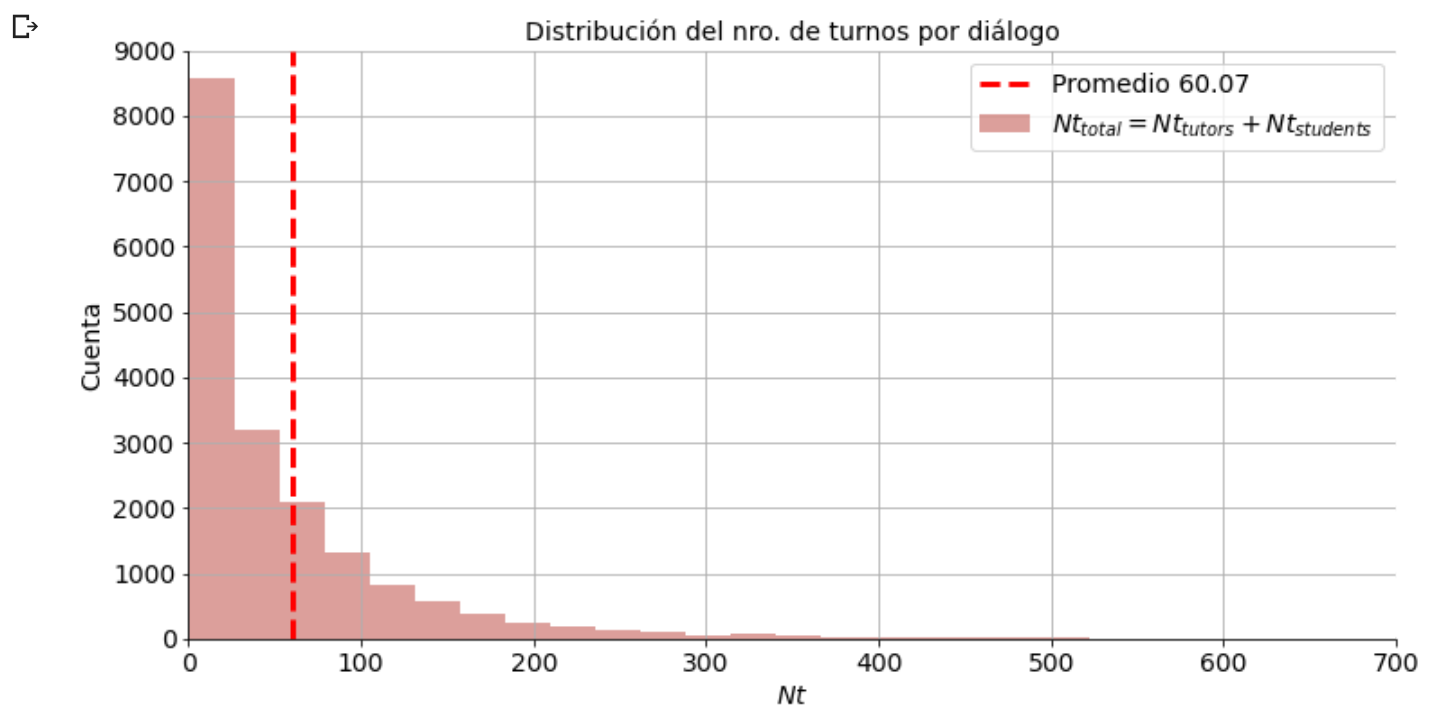
El Promedio de turnos por tutores de 36.22, mientras que el Promedio de turnos por estudiantes es 60.07'

▼ Histograma del nro. total de turnos por diálogo

$$Nt_{total} = Nt_{tutors} + Nt_{students}$$

donde  $Nt$  es el número total de turnos por diálogo, teniendo en cuenta los turnos por tutor y por estudiantes.

```
1 plt.figure(figsize=(12,6))
2 plt.plot([mean_total, mean_total], [0,9000], '--r', linewidth=3, label=f'Promedio {mean_total:.2f}')
3 sns.distplot(turnos.counts.dropna(), kde=False, label=r'$Nt_{total} = Nt_{tutors} + Nt_{students}$')
4 plt.grid(True)
5 plt.title("Distribución del nro. de turnos por diálogo", fontsize=14)
6 plt.xlabel(r'$Nt$', fontsize=14)
7 plt.ylabel('Cuenta', fontsize=14)
8 plt.xticks(fontsize=14)
9 plt.yticks(fontsize=14)
10 plt.legend(fontsize=14)
11 plt.xlim([0,700])
12 plt.ylim([0, 9000])
13 sns.despine()
```

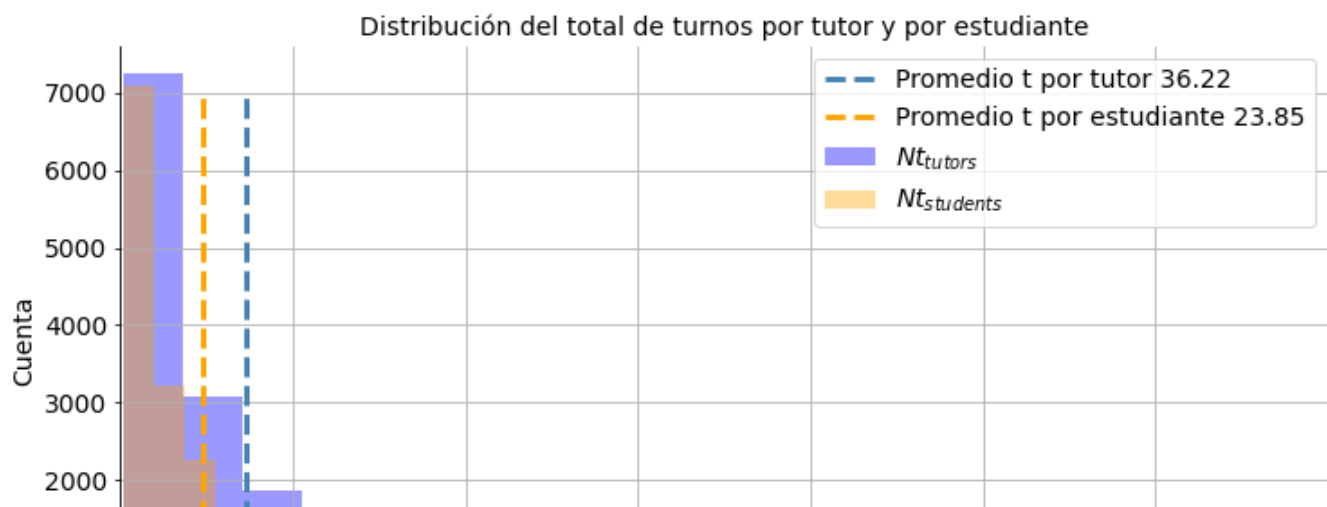


En el anterior histograma se puede observar que la frecuencia disminuye a medida que aumenta el nro. de turnos y que los valores máximos se concentran a la izquierda del histograma. Además, se puede afirmar que predominan los diálogos cortos.

▼ Histograma de turnos discriminado por estudiantes y tutores

```
1 plt.figure(figsize=(12,6))
2 sns.distplot(st_turnos[st_turnos.sent_from=='tutor'].counts.dropna(), kde=False, label=r'$Nt_{tutor}$')
3 sns.distplot(st_turnos[st_turnos.sent_from=='student'].counts.dropna(), kde=False, label=r'$Nt_{student}$')
4 plt.plot([mean_tutors, mean_tutors], [0,7050], '--b', linewidth=3, label=f'Promedio t por tutor {mean_tutors:.2f}')
5 plt.plot([mean_students, mean_students], [0,7050], '--g', linewidth=3, label=f'Promedio t por estudiante {mean_students:.2f}')
6 plt.grid(True)
7 plt.title("Distribución del total de turnos por tutor y por estudiante", fontsize=14)
8 plt.xlabel(r'$Nt$', fontsize=14)
9 plt.ylabel('Cuenta', fontsize=14)
10 plt.xticks(fontsize=14)
11 plt.yticks(fontsize=14)
12 plt.legend(fontsize=14)
13 plt.xlim([0,350])
14 #plt.ylim([0, 55])
15 sns.despine()
```





En el anterior histograma se evidencia que el promedio del nro. de turnos del tutor es mayor que el del estudiante, existiendo una diferencia de aprox. 12 turnos. Sin embargo, se evidencia en ambos histogramas que los valores máximo se presentan en un extremo del mismo y que la distribución es inversa ya que a medida que aumenta el nro. de turnos por diálogo disminuye la frecuencia de los mismos.

## Conclusión

- Un diálogo entre un tutor y un estudiante está formado aproximadamente por un promedio de 60 turnos.
- Los casos con más de 200 turnos son poco probales de acuerdo a los cálculos y el histograma gráficoado.

Un aspecto de interés es la cantidad total de turnos discriminada entre estudiantes y tutores. Esto permite comparar la proporcion de participación de de ambos en el desarrollo de un diálogo, evidenciándose, en términos generales, que existe mayor participación del tutor en los diálogos.