

INFORME DEL PROYECTO DE FUNDAMENTOS DE CIENCIA DE DATOS

Laura Daniela Rojas Ardila - U00144894
Stiven Alexis Villamizar Buitrago - U00157009

CÓDIGO

IMPORTE DE LIBRERÍAS

```
def get_player_puuid(game_name, tag_line, api_key):
    """Obtiene el identificador único (PUUID) del jugador a
    partir de su Riot ID."""
    request_url =
f"https://americas.api.riotgames.com/riot/account/v1/accounts
/by-riot-id/{game_name}/{tag_line}?api_key={api_key}"
    respuesta = requests.get(request_url)
    if respuesta.status_code == 200:
        return respuesta.json().get('puuid')
    else:
        print("Error fetching player data: ",
respuesta.status_code)
        return None

def get_matches(region, puuid, count, api_key):
    """Obtiene la lista de IDs de partidas jugadas por el
    usuario."""
    api_url =
f"https://{region}.api.riotgames.com/lol/match/v5/matches/by-
puuid/{puuid}/ids?type=ranked&start=0&count={count}&api_key={
api_key}"
    respuesta = requests.get(api_url)
    return respuesta.json() if respuesta.status_code == 200
    else []

def get_match_data(region, match_id, api_key):
    """Obtiene los datos detallados de una partida
    específica."""
    api_url =
f"https://{region}.api.riotgames.com/lol/match/v5/matches/{ma
tch_id}?api_key={api_key}"
    while True:
        respuesta = requests.get(api_url)
        if respuesta.status_code == 429:
            print("Rate limit exceeded. Sleeping for 10
seconds")
            time.sleep(10)
            continue
        elif respuesta.status_code != 200:
```

```

        print(f"Error fetching match data for {match_id}: {respuesta.status_code}")
        return None
    return respuesta.json()

```

```

def collect_match_data(region, puuid, matches, api_key):
    """Recopila los datos relevantes de múltiples partidas y los almacena en un DataFrame."""
    match_results = {}
    for idx, match_id in enumerate(matches, start=1):
        print(f"Fetching data for match {idx}: {match_id}")
        match_data = get_match_data(region, match_id, api_key)
        if not match_data:
            continue
        try:
            part_index =
match_data['metadata']['participants'].index(puuid)
            participant_data =
match_data['info']['participants'][part_index]
        except ValueError:
            print(f"Player not found in match {match_id}")
            continue
        match_results[match_id] =
extract_match_features(match_data, participant_data)
        time.sleep(2.5)
    return pd.DataFrame.from_dict(match_results,
orient='index')

```

```

def extract_match_features(match_data, participant_data):
    """Extrae características clave de una partida y del jugador en cuestión."""
    return {
        'win': participant_data['win'],
        'championName': participant_data['championName'],
        'kills': participant_data['kills'],
        'deaths': participant_data['deaths'],
        'assists': participant_data['assists'],
        'gameDuration':
match_data['info'].get('gameDuration'),
        'totalMinionsKilled':
participant_data.get('totalMinionsKilled'),
        'totalDamageDealtToChampions':
participant_data.get('totalDamageDealtToChampions'),
        'wardsPlaced': participant_data.get('wardsPlaced'),
        'wardsDestroyed':
participant_data.get('wardsDestroyed'),
        'dragonKills': participant_data.get('dragonKills'),
        'baronKills': participant_data.get('baronKills'),
        'turretTakedowns':
participant_data.get('turretTakedowns'),

```

```

        'firstBloodKill':
participant_data.get('firstBloodKill'),
        'firstTowerKill':
participant_data.get('firstTowerKill'),
        'firstBaron': participant_data.get('firstBaron'),
        'firstDragon': participant_data.get('firstDragon'),
    }

```

```

def analyze_data(df):
    """Realiza análisis estadístico y visualización de los
    datos."""
    numeric_df = df.select_dtypes(include=[np.number])
    print(numeric_df.describe())
    correlation_matrix = df.corr(numeric_only=True)
    plt.figure(figsize=(10, 8))
    sns.heatmap(correlation_matrix, annot=True,
cmap="coolwarm", fmt=".2f", linewidths=0.5)
    plt.title("Matriz de Correlación")
    plt.show()

    stats = numeric_df.describe().T
    for column in numeric_df.columns:
        plt.figure(figsize=(8, 5))
        plt.hist(numeric_df[column], bins=20, color='skyblue',
edgecolor='black')
        plt.title(f'Distribución de {column}')
        plt.xlabel(column)
        plt.ylabel('Frecuencia')
        plt.grid(True)
        plt.show()

        plt.figure(figsize=(6, 4))
        sns.boxplot(x=numeric_df[column])
        plt.title(f'Boxplot de {column}')
        plt.grid(True)
        plt.show()

    return stats

```

```

def train_model(df):
    """Entrena un modelo de Random Forest para predecir
    victorias en partidas."""
    df.fillna(0, inplace=True)
    x = df.drop(['win', 'championName'], axis=1)
    y = df['win']
    x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.2)
    clf = RandomForestClassifier(n_jobs=-1)
    clf.fit(x_train, y_train)
    accuracy = clf.score(x_test, y_test)
    print("Model accuracy: ", accuracy * 100)

```

```

    return clf, accuracy

def save_model(clf, df, accuracy):
    """Guarda el modelo entrenado y los datos si la precisión
    es suficiente."""
    if accuracy >= 0.75:
        os.makedirs("modelos_guardados", exist_ok=True)
        model_filename =
f"modelos_guardados/modelo_{int(accuracy * 100)}.pkl"
        data_filename =
f"modelos_guardados/datos_{int(accuracy * 100)}.csv"
        joblib.dump(clf, model_filename)
        df.to_csv(data_filename, index=False)
        print(f"✓ Modelo guardado como {model_filename}")
        print(f"✓ Datos guardados como {data_filename}")

```

LLAMADO DEL FLUJO DEL PROGRAMA Y RESULTADOS

```

game_name, tag_line, region = "Rerzd", "LAN", "americas"
api_key = "RGAPI-3bb3e6a8-f965-4229-914f-4dc622183e8e"
match_count = 20

```

```

puuid = get_player_puuid(game_name, tag_line, api_key)

```

Resultado que se genera:

```

{'puuid':
'W_RelwbtQyoQrMAg0o5MobSlidnnbVTGU0p1Ot2nQ5YUz8HARYWRzgqElc5
gJbaWqTDUdM7DqZ4ztg', 'gameName': 'Rerzd', 'tagLine': 'LAN'}
W_RelwbtQyoQrMAg0o5MobSlidnnbVTGU0p1Ot2nQ5YUz8HARYWRzgqElc5
gJbaWqTDUdM7DqZ4ztg

```

```

matches = get_matches(region, puuid, match_count, api_key)

```

Enlace necesario para el funcionamiento de la función:

https://americas.api.riotgames.com/lol/match/v5/matches/by-puuid/W_RelwbtQyoQrMAg0o5MobSlidnnbVTGU0p1Ot2nQ5YUz8HARYWRzgqElc5gJbaWqTDUdM7DqZ4ztg/ids?type=ranked&start=0&count=20&api_key=RGAPI-13dc4d8a-2a5b-49ad-b07c-c2d5b4ba080b

```

df = collect_match_data(region, puuid, matches, api_key)

```

Resultados obtenidos:

Fetching data for match 1: LA1_1602634235

Fetching data for match 2: LA1_1600995560
 Fetching data for match 3: LA1_1598294323
 Fetching data for match 4: LA1_1554728484
 Fetching data for match 5: LA1_1554712970
 Fetching data for match 6: LA1_1554708544
 Fetching data for match 7: LA1_1554687978
 Fetching data for match 8: LA1_1553034414
 Fetching data for match 9: LA1_1553016163
 Fetching data for match 10: LA1_1552990950
 Fetching data for match 11: LA1_1534781476
 Fetching data for match 12: LA1_1534729048
 Fetching data for match 13: LA1_1534624814
 Fetching data for match 14: LA1_1534607036
 Fetching data for match 15: LA1_1534563171
 Fetching data for match 16: LA1_1534540125
 Fetching data for match 17: LA1_1534109175
 Fetching data for match 18: LA1_1534049033
 Fetching data for match 19: LA1_1534001367
 Fetching data for match 20: LA1_1533936605

Tablas Que Se Crean A Partir De Los Datos:

| | win | championName | kills | deaths | assists | gameDuration | \ |
|----------------|-------|--------------|-------|--------|---------|--------------|---|
| LA1_1602634235 | True | Rengar | 6 | 0 | 4 | 1182 | |
| LA1_1600995560 | True | Lillia | 6 | 4 | 9 | 1518 | |
| LA1_1598294323 | False | Rengar | 3 | 7 | 3 | 1537 | |
| LA1_1554728484 | False | Garen | 7 | 6 | 5 | 1928 | |
| LA1_1554712970 | True | Garen | 12 | 6 | 8 | 2005 | |
| LA1_1554708544 | False | Rengar | 1 | 5 | 2 | 911 | |
| LA1_1554687978 | False | Garen | 9 | 7 | 5 | 2511 | |
| LA1_1553034414 | True | Garen | 10 | 1 | 5 | 1712 | |
| LA1_1553016163 | False | Rengar | 6 | 10 | 16 | 2196 | |
| LA1_1552990950 | False | Sejuani | 5 | 10 | 6 | 1434 | |
| LA1_1534781476 | False | Rengar | 13 | 7 | 6 | 1777 | |
| LA1_1534729048 | False | Mordekaiser | 7 | 9 | 7 | 2413 | |
| LA1_1534624814 | False | Garen | 3 | 6 | 4 | 1451 | |
| LA1_1534607036 | False | Pyke | 1 | 4 | 4 | 952 | |
| LA1_1534563171 | True | Morgana | 2 | 8 | 21 | 2282 | |
| LA1_1534540125 | False | Rengar | 3 | 5 | 0 | 937 | |
| LA1_1534109175 | False | Rengar | 10 | 7 | 8 | 1644 | |
| LA1_1534049033 | True | Rengar | 5 | 9 | 15 | 2611 | |
| LA1_1534001367 | True | Rengar | 11 | 3 | 6 | 1641 | |
| LA1_1533936605 | False | Lillia | 8 | 7 | 8 | 2291 | |

| | totalMinionsKilled | totalDamageDealtToChampions | wardsPlaced | \ |
|----------------|--------------------|-----------------------------|-------------|---|
| LA1_1602634235 | 26 | 9588 | 2 | |
| LA1_1600995560 | 15 | 12707 | 9 | |
| LA1_1598294323 | 16 | 8866 | 1 | |
| LA1_1554728484 | 198 | 36231 | 14 | |
| LA1_1554712970 | 228 | 33200 | 13 | |
| LA1_1554708544 | 11 | 6073 | 5 | |
| LA1_1554687978 | 281 | 28547 | 17 | |
| LA1_1553034414 | 171 | 22425 | 8 | |
| LA1_1553016163 | 13 | 23083 | 6 | |
| LA1_1552990950 | 15 | 12351 | 2 | |
| LA1_1534781476 | 31 | 25519 | 2 | |
| LA1_1534729048 | 197 | 34649 | 3 | |
| LA1_1534624814 | 127 | 20542 | 8 | |
| LA1_1534607036 | 70 | 3747 | 10 | |
| LA1_1534563171 | 59 | 20755 | 27 | |
| LA1_1534540125 | 8 | 6020 | 6 | |
| LA1_1534109175 | 37 | 17446 | 9 | |
| LA1_1534049033 | 44 | 24278 | 2 | |
| LA1_1534001367 | 43 | 18893 | 2 | |
| LA1_1533936605 | 40 | 25976 | 1 | |

| | wardsDestroyed | dragonKills | baronKills | turretTakedowns | \ |
|----------------|----------------|-------------|------------|-----------------|---|
| LA1_1602634235 | None | 2 | 0 | 0 | |
| LA1_1600995560 | None | 3 | 0 | 1 | |
| LA1_1598294323 | None | 1 | 0 | 0 | |
| LA1_1554728484 | None | 0 | 0 | 1 | |
| LA1_1554712970 | None | 0 | 0 | 5 | |
| LA1_1554708544 | None | 1 | 0 | 0 | |
| LA1_1554687978 | None | 1 | 0 | 5 | |
| LA1_1553034414 | None | 0 | 0 | 2 | |
| LA1_1553016163 | None | 1 | 1 | 0 | |
| LA1_1552990950 | None | 0 | 0 | 0 | |
| LA1_1534781476 | None | 0 | 0 | 0 | |
| LA1_1534729048 | None | 0 | 0 | 2 | |
| LA1_1534624814 | None | 0 | 0 | 3 | |
| LA1_1534607036 | None | 0 | 0 | 0 | |
| LA1_1534563171 | None | 0 | 0 | 1 | |
| LA1_1534540125 | None | 0 | 0 | 0 | |
| LA1_1534109175 | None | 1 | 0 | 0 | |
| LA1_1534049033 | None | 2 | 0 | 3 | |
| LA1_1534001367 | None | 1 | 0 | 0 | |
| LA1_1533936605 | None | 1 | 0 | 0 | |

| | firstBloodKill | firstTowerKill | firstBaron | firstDragon |
|----------------|----------------|----------------|------------|-------------|
| LA1_1602634235 | False | False | None | None |
| LA1_1600995560 | False | False | None | None |
| LA1_1598294323 | False | False | None | None |
| LA1_1554728484 | False | False | None | None |
| LA1_1554712970 | False | False | None | None |
| LA1_1554708544 | False | False | None | None |
| LA1_1554687978 | False | False | None | None |
| LA1_1553034414 | False | True | None | None |
| LA1_1553016163 | True | False | None | None |
| LA1_1552990950 | False | False | None | None |
| LA1_1534781476 | False | False | None | None |
| LA1_1534729048 | False | True | None | None |
| LA1_1534624814 | False | False | None | None |
| LA1_1534607036 | False | False | None | None |
| LA1_1534563171 | False | False | None | None |
| LA1_1534540125 | False | False | None | None |
| LA1_1534109175 | False | False | None | None |
| LA1_1534049033 | False | False | None | None |
| LA1_1534001367 | False | False | None | None |
| LA1_1533936605 | True | False | None | None |

analyze_data(df)

Tablas Que Se Crean A Partir De Los Datos:

Descriptive Statistics For Numeric Columns:

| | kills | deaths | assists | gameDuration | totalMinionsKilled | \ |
|-------|-----------|-----------|---------|--------------|--------------------|---|
| count | 20.000000 | 20.000000 | 20.0000 | 20.000000 | 20.00000 | |
| mean | 6.400000 | 6.050000 | 7.1000 | 1746.650000 | 81.50000 | |
| std | 3.618665 | 2.704285 | 5.0252 | 526.616536 | 85.62925 | |
| min | 1.000000 | 0.000000 | 0.0000 | 911.000000 | 8.00000 | |
| 25% | 3.000000 | 4.750000 | 4.0000 | 1446.750000 | 15.75000 | |
| 50% | 6.000000 | 6.500000 | 6.0000 | 1678.000000 | 41.50000 | |
| 75% | 9.250000 | 7.250000 | 8.0000 | 2217.500000 | 138.00000 | |
| max | 13.000000 | 10.000000 | 21.0000 | 2611.000000 | 281.00000 | |

| | totalDamageDealtToChampions | wardsPlaced | dragonKills | baronKills |
|-------|-----------------------------|-------------|-------------|------------|
| count | 20.000000 | 20.00000 | 20.000000 | 20.000000 |
| mean | 19544.800000 | 7.35000 | 0.700000 | 0.050000 |
| std | 9806.788753 | 6.57167 | 0.864505 | 0.223607 |
| min | 3747.000000 | 1.00000 | 0.000000 | 0.000000 |
| 25% | 11660.250000 | 2.00000 | 0.000000 | 0.000000 |
| 50% | 20648.500000 | 6.00000 | 0.500000 | 0.000000 |
| 75% | 25633.250000 | 9.25000 | 1.000000 | 0.000000 |
| max | 36231.000000 | 27.00000 | 3.000000 | 1.000000 |

| | turretTakedowns |
|-------|-----------------|
| count | 20.000000 |
| mean | 1.150000 |
| std | 1.663066 |
| min | 0.000000 |
| 25% | 0.000000 |
| 50% | 0.000000 |
| 75% | 2.000000 |
| max | 5.000000 |

Descriptive statistics for champion stats:

| | matches | wins | Avg_kills | Avg_deaths | Avg_assists | \ |
|-------|----------|----------|-----------|------------|-------------|---|
| count | 7.000000 | 7.000000 | 7.000000 | 7.000000 | 7.000000 | |
| mean | 2.857143 | 1.000000 | 5.234921 | 6.798413 | 8.366667 | |
| std | 3.078342 | 1.154701 | 2.736790 | 2.215104 | 5.742338 | |
| min | 1.000000 | 0.000000 | 1.000000 | 4.000000 | 4.000000 | |
| 25% | 1.000000 | 0.000000 | 3.500000 | 5.350000 | 5.700000 | |
| 50% | 1.000000 | 1.000000 | 6.444444 | 5.888889 | 6.666667 | |
| 75% | 3.500000 | 1.500000 | 7.000000 | 8.500000 | 7.750000 | |
| max | 9.000000 | 3.000000 | 8.200000 | 10.000000 | 21.000000 | |

| | Avg_gameDuration | Avg_MinionsKilled | Avg_DamageDealtToChampions | \ |
|-------|------------------|-------------------|----------------------------|---|
| count | 7.000000 | 7.000000 | 7.000000 | |
| mean | 1787.271429 | 84.992063 | 19223.150794 | |
| std | 503.999520 | 80.264265 | 10174.184570 | |
| min | 952.000000 | 15.000000 | 3747.000000 | |
| 25% | 1519.000000 | 26.472222 | 13940.277778 | |
| 50% | 1904.500000 | 59.000000 | 19341.500000 | |
| 75% | 2101.700000 | 133.500000 | 24472.000000 | |
| max | 2413.000000 | 201.000000 | 34649.000000 | |

| | Avg_wardsPlaced | Avg_dragonKills | Avg_baronKills | Avg_turretTakedowns | \ |
|-------|-----------------|-----------------|----------------|---------------------|---|
| count | 7.000000 | 7.000000 | 7.000000 | 7.000000 | |
| mean | 8.984127 | 0.457143 | 0.015873 | 1.004762 | |
| std | 8.766619 | 0.772134 | 0.041996 | 1.192636 | |
| min | 2.000000 | 0.000000 | 0.000000 | 0.000000 | |
| 25% | 3.444444 | 0.000000 | 0.000000 | 0.166667 | |
| 50% | 5.000000 | 0.000000 | 0.000000 | 0.500000 | |
| 75% | 11.000000 | 0.600000 | 0.000000 | 1.500000 | |
| max | 27.000000 | 2.000000 | 0.111111 | 3.200000 | |

| | Avg_firstBloodKill | Avg_firstTowerKill | win_rate |
|-------|--------------------|--------------------|------------|
| count | 7.000000 | 7.000000 | 7.000000 |
| mean | 0.087302 | 0.171429 | 31.904762 |
| std | 0.186635 | 0.372891 | 36.709931 |
| min | 0.000000 | 0.000000 | 0.000000 |
| 25% | 0.000000 | 0.000000 | 0.000000 |
| 50% | 0.000000 | 0.000000 | 33.333333 |
| 75% | 0.055556 | 0.100000 | 45.000000 |
| max | 0.500000 | 1.000000 | 100.000000 |

Descriptive statistics aggregated by championName:

| | matches | wins | Avg_kills | Avg_deaths | Avg_assists \ |
|--------------|---------|------|-----------|------------|---------------|
| championName | | | | | |
| Garen | 5 | 2 | 8.200000 | 5.200000 | 5.400000 |
| Lillia | 2 | 1 | 7.000000 | 5.500000 | 8.500000 |
| Mordekaiser | 1 | 0 | 7.000000 | 9.000000 | 7.000000 |
| Morgana | 1 | 1 | 2.000000 | 8.000000 | 21.000000 |
| Pyke | 1 | 0 | 1.000000 | 4.000000 | 4.000000 |
| Rengar | 9 | 3 | 6.444444 | 5.888889 | 6.666667 |
| Sejuani | 1 | 0 | 5.000000 | 10.000000 | 6.000000 |

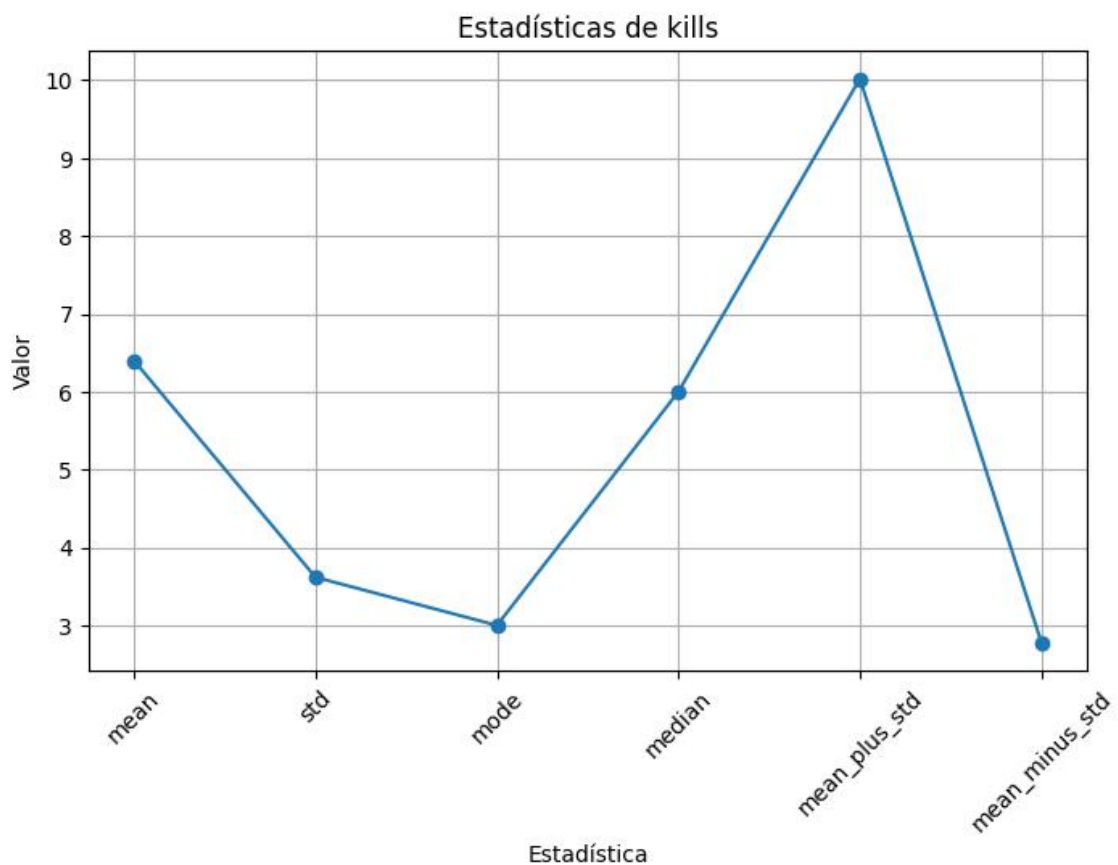
| | Avg_gameDuration | Avg_MinionsKilled | Avg_DamageDealtToChampions \ |
|--------------|------------------|-------------------|------------------------------|
| championName | | | |
| Garen | 1921.4 | 201.000000 | 28189.000000 |
| Lillia | 1904.5 | 27.500000 | 19341.500000 |
| Mordekaiser | 2413.0 | 197.000000 | 34649.000000 |
| Morgana | 2282.0 | 59.000000 | 20755.000000 |
| Pyke | 952.0 | 70.000000 | 3747.000000 |
| Rengar | 1604.0 | 25.444444 | 15529.555556 |
| Sejuani | 1434.0 | 15.000000 | 12351.000000 |

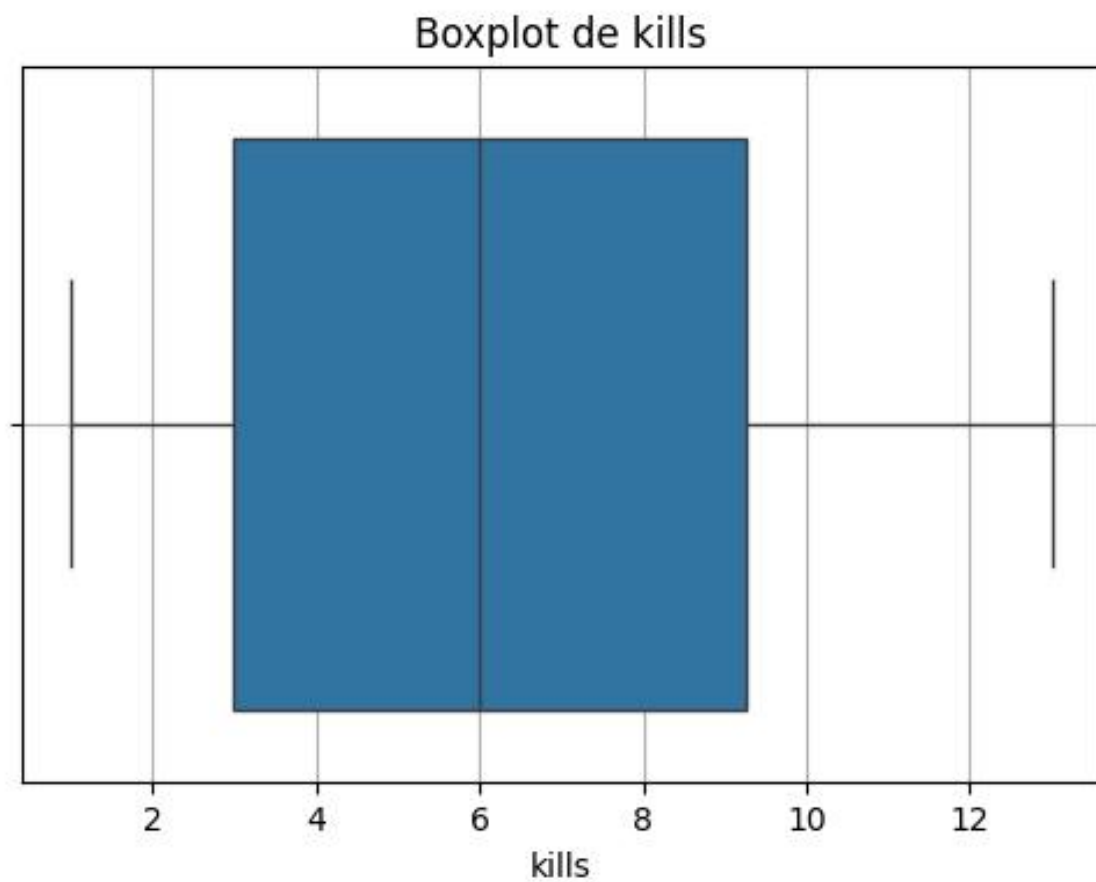
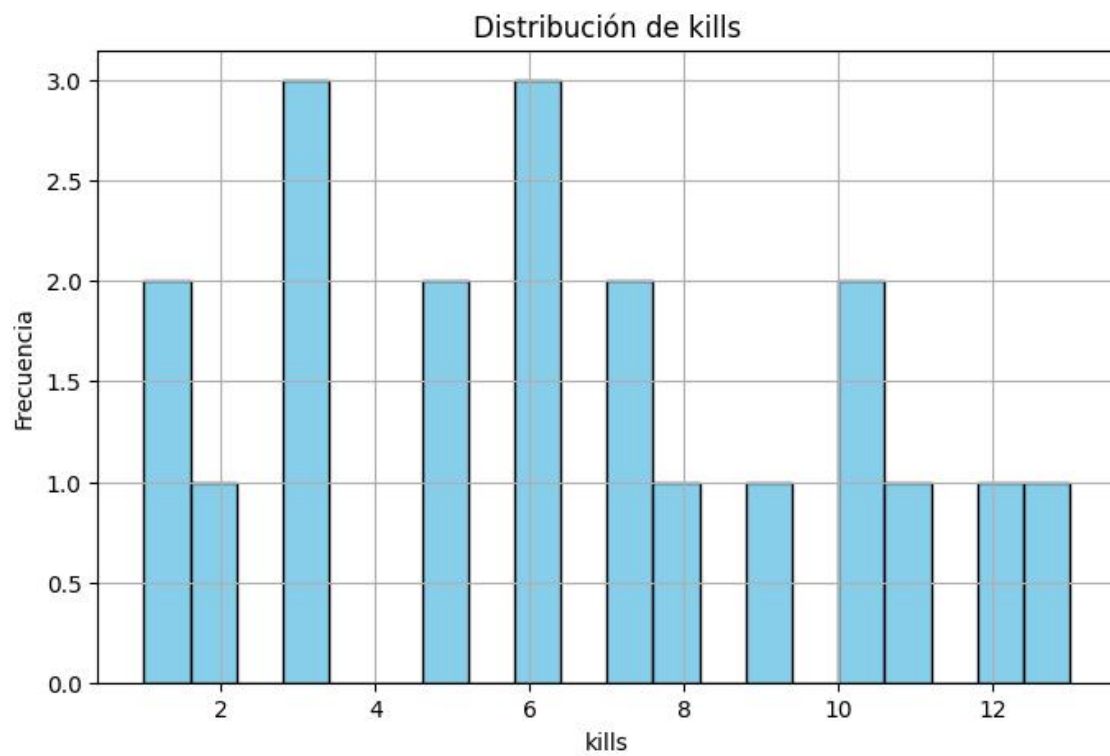
| | Avg_wardsPlaced | Avg_wardsDestroyed | Avg_dragonKills \ |
|--------------|-----------------|--------------------|-------------------|
| championName | | | |
| Garen | 12.000000 | 0 | 0.2 |
| Lillia | 5.000000 | 0 | 2.0 |
| Mordekaiser | 3.000000 | 0 | 0.0 |
| Morgana | 27.000000 | 0 | 0.0 |
| Pyke | 10.000000 | 0 | 0.0 |
| Rengar | 3.888889 | 0 | 1.0 |
| Sejuani | 2.000000 | 0 | 0.0 |

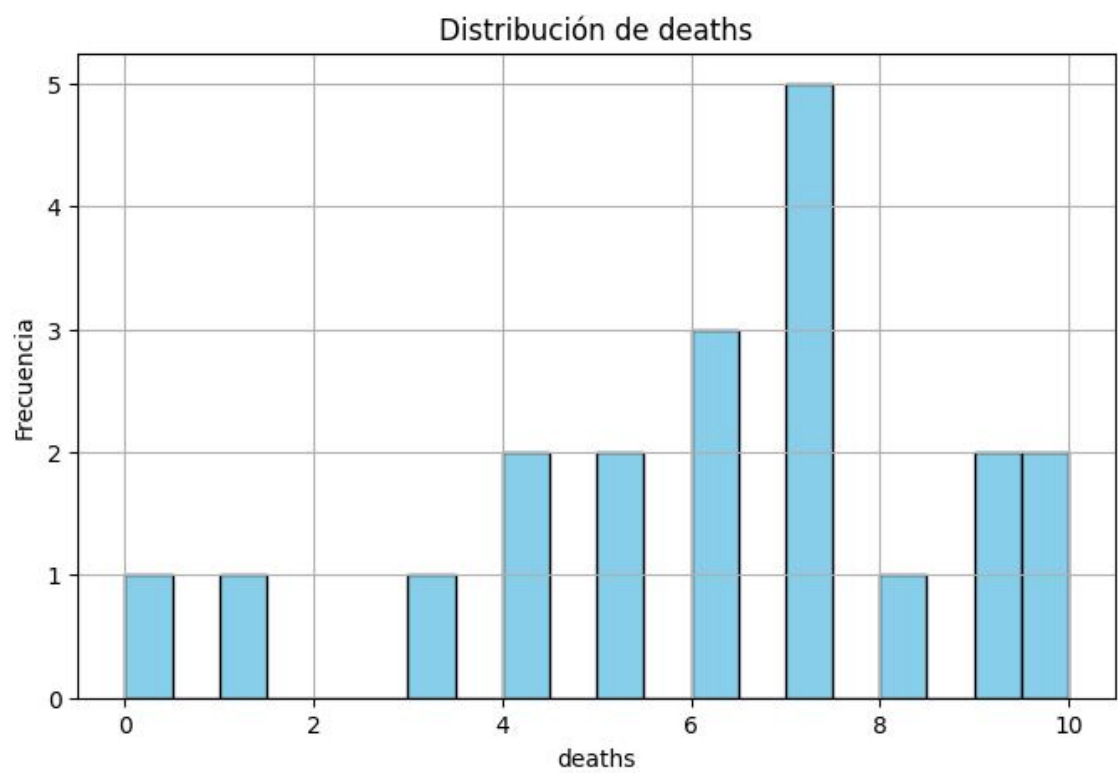
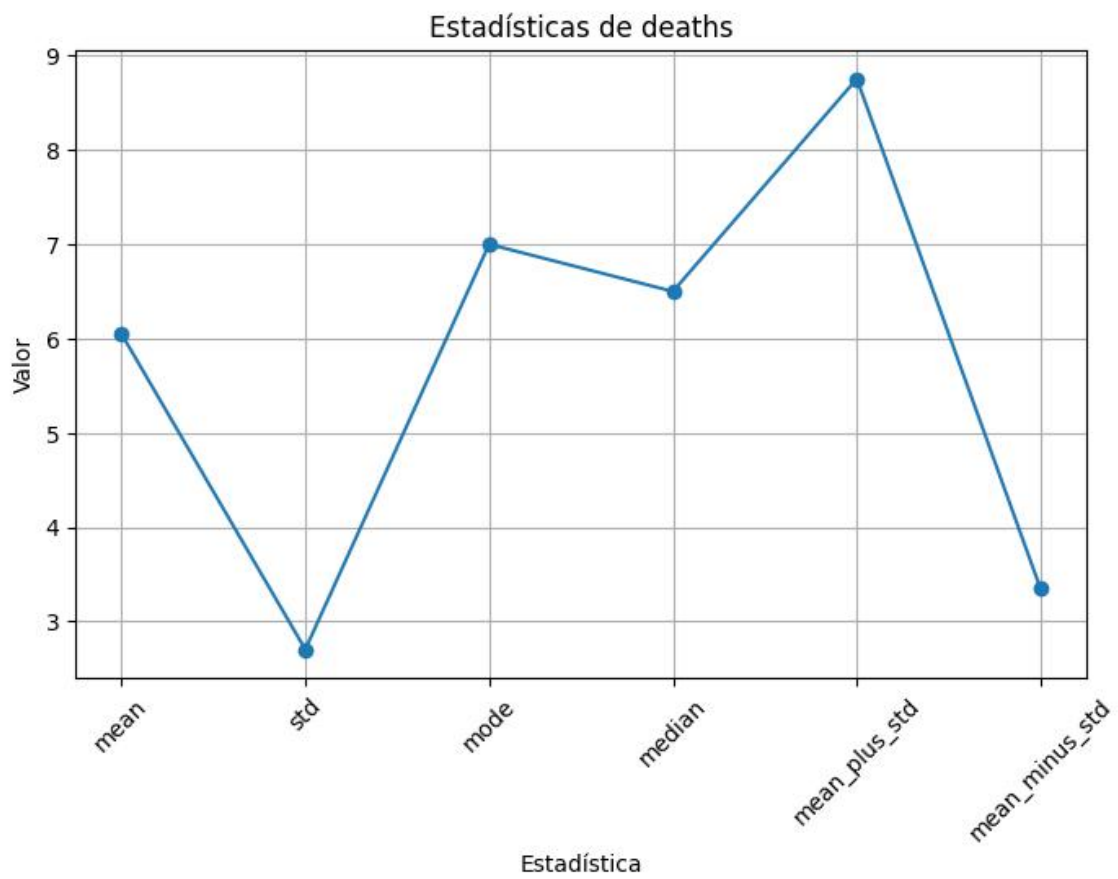
| championName | Avg_baronKills | Avg_turretTakedowns | Avg_firstBloodKill | \ |
|--------------|----------------|---------------------|--------------------|---|
| Garen | 0.000000 | 3.200000 | 0.000000 | |
| Lillia | 0.000000 | 0.500000 | 0.500000 | |
| Mordekaiser | 0.000000 | 2.000000 | 0.000000 | |
| Morgana | 0.000000 | 1.000000 | 0.000000 | |
| Pyke | 0.000000 | 0.000000 | 0.000000 | |
| Rengar | 0.111111 | 0.333333 | 0.111111 | |
| Sejuani | 0.000000 | 0.000000 | 0.000000 | |

| championName | Avg_firstTowerKill | Avg_firstBaron | Avg_firstDragon | win_rate |
|--------------|--------------------|----------------|-----------------|------------|
| Garen | 0.2 | 0 | 0 | 40.000000 |
| Lillia | 0.0 | 0 | 0 | 50.000000 |
| Mordekaiser | 1.0 | 0 | 0 | 0.000000 |
| Morgana | 0.0 | 0 | 0 | 100.000000 |
| Pyke | 0.0 | 0 | 0 | 0.000000 |
| Rengar | 0.0 | 0 | 0 | 33.333333 |
| Sejuani | 0.0 | 0 | 0 | 0.000000 |

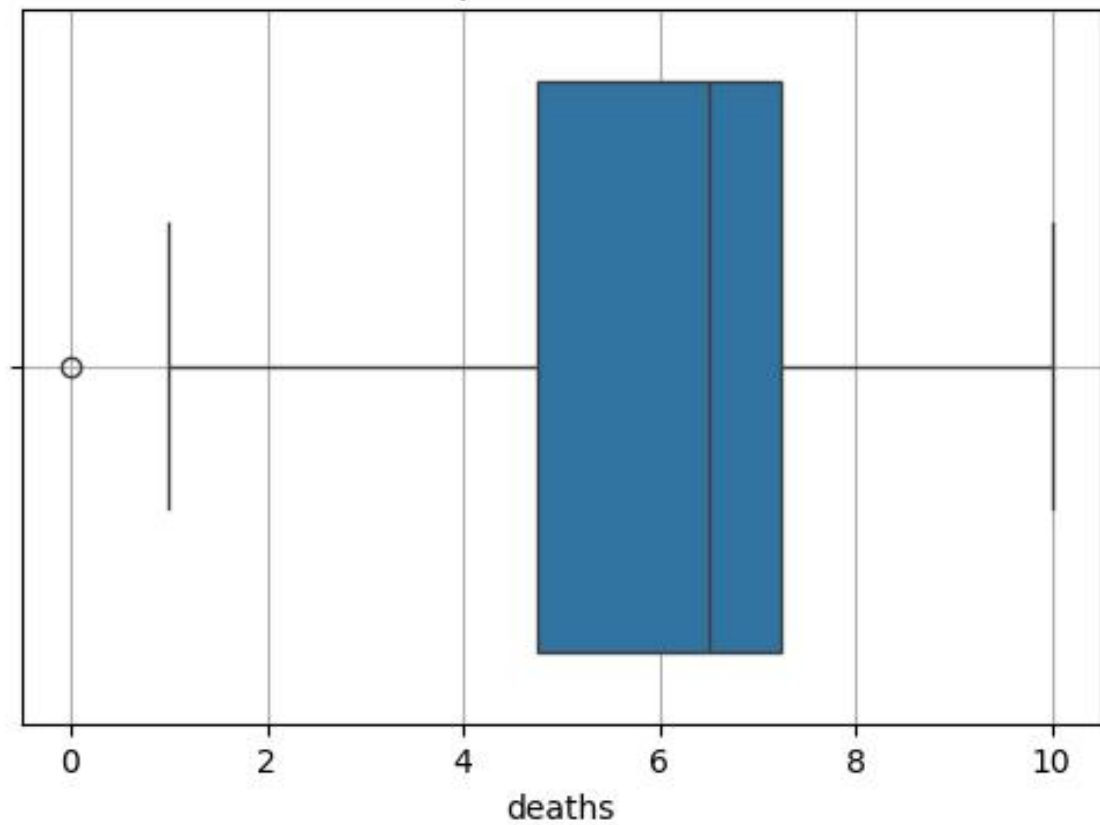
Gráficas De Los Resultados Estadísticos



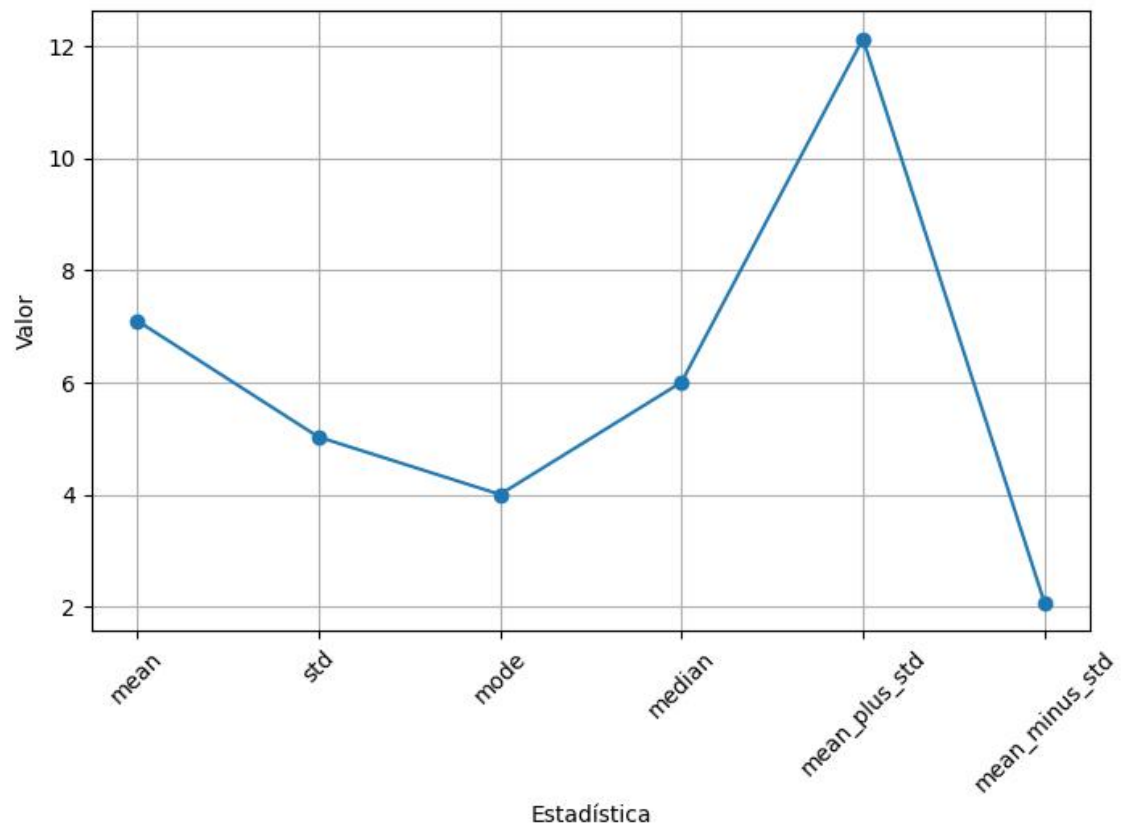


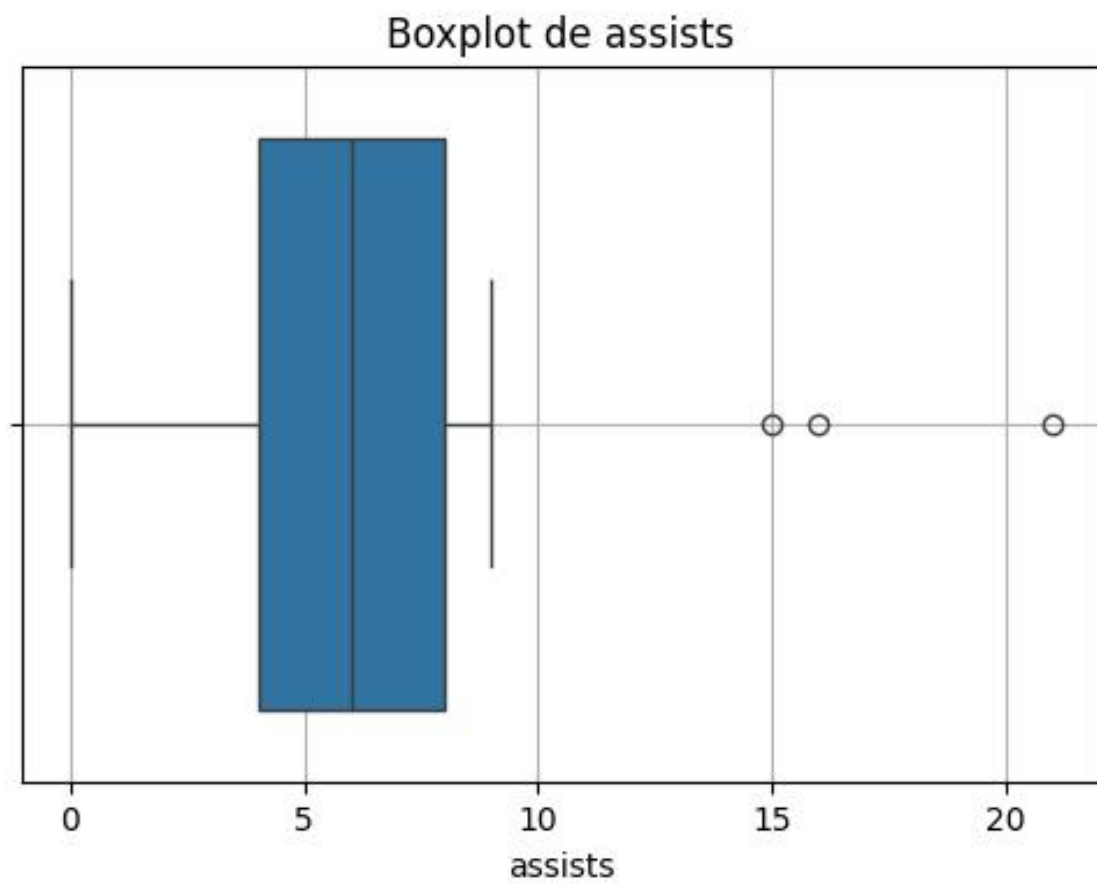
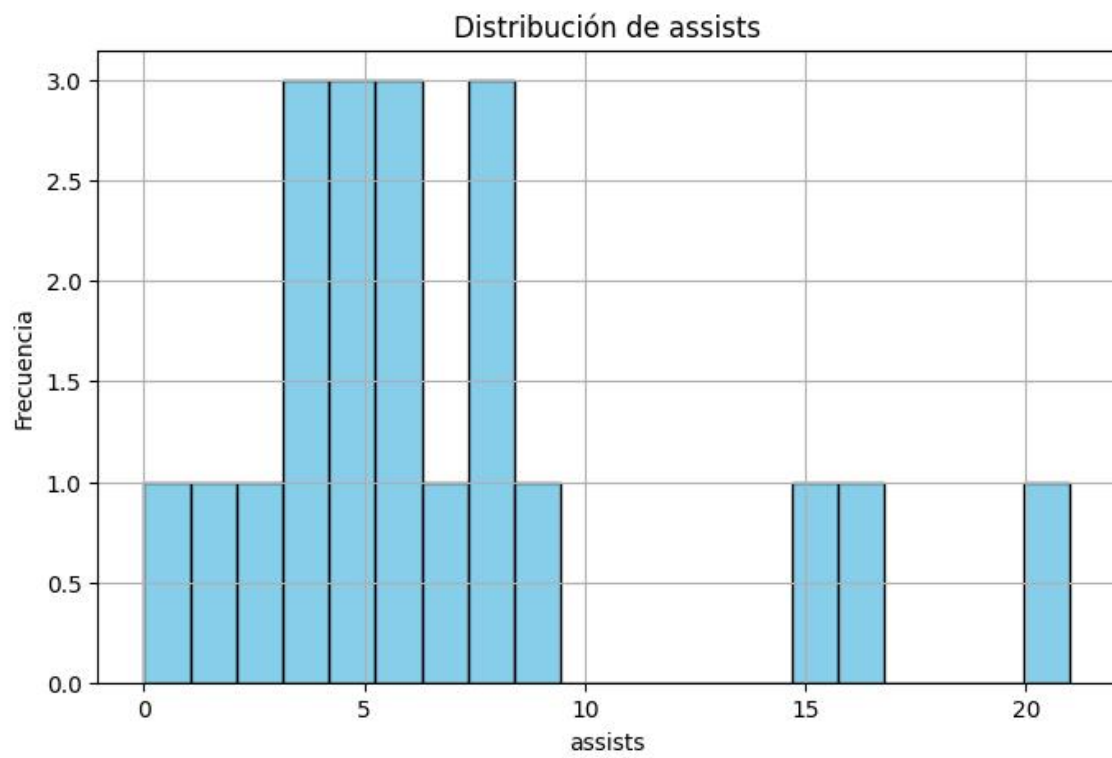


Boxplot de deaths



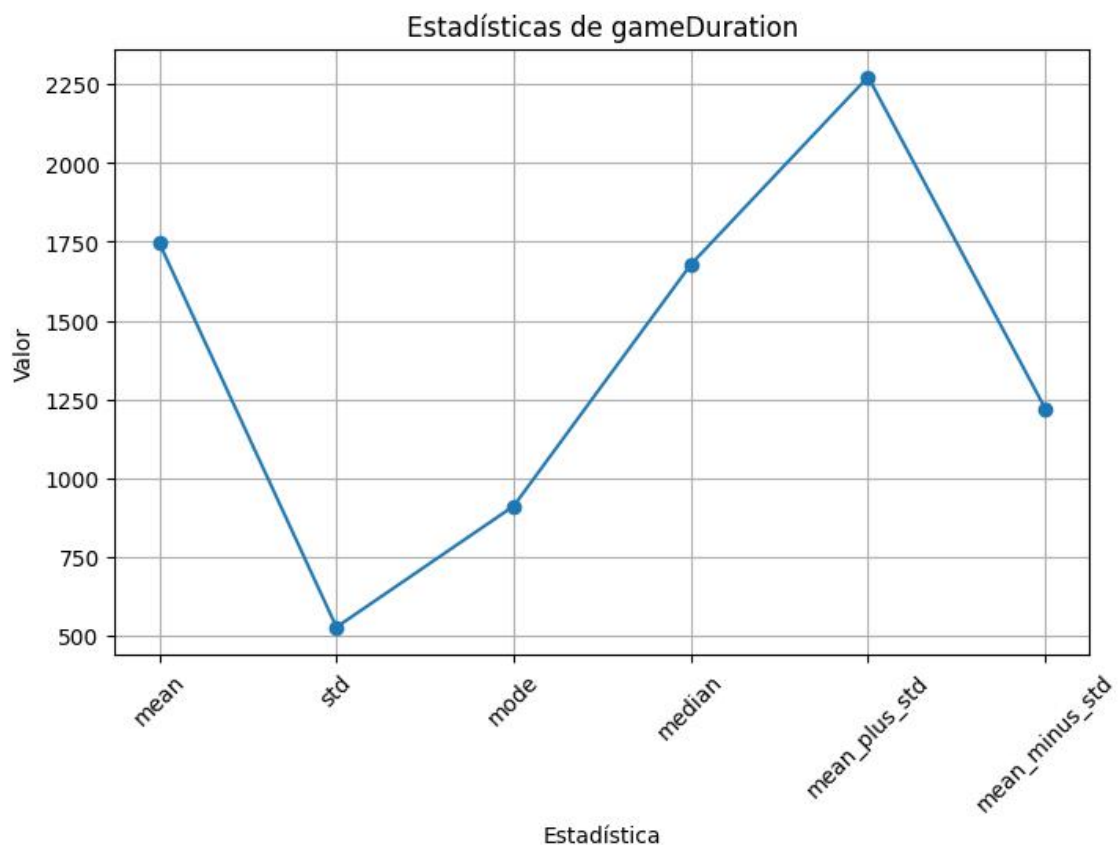
Estadísticas de assists

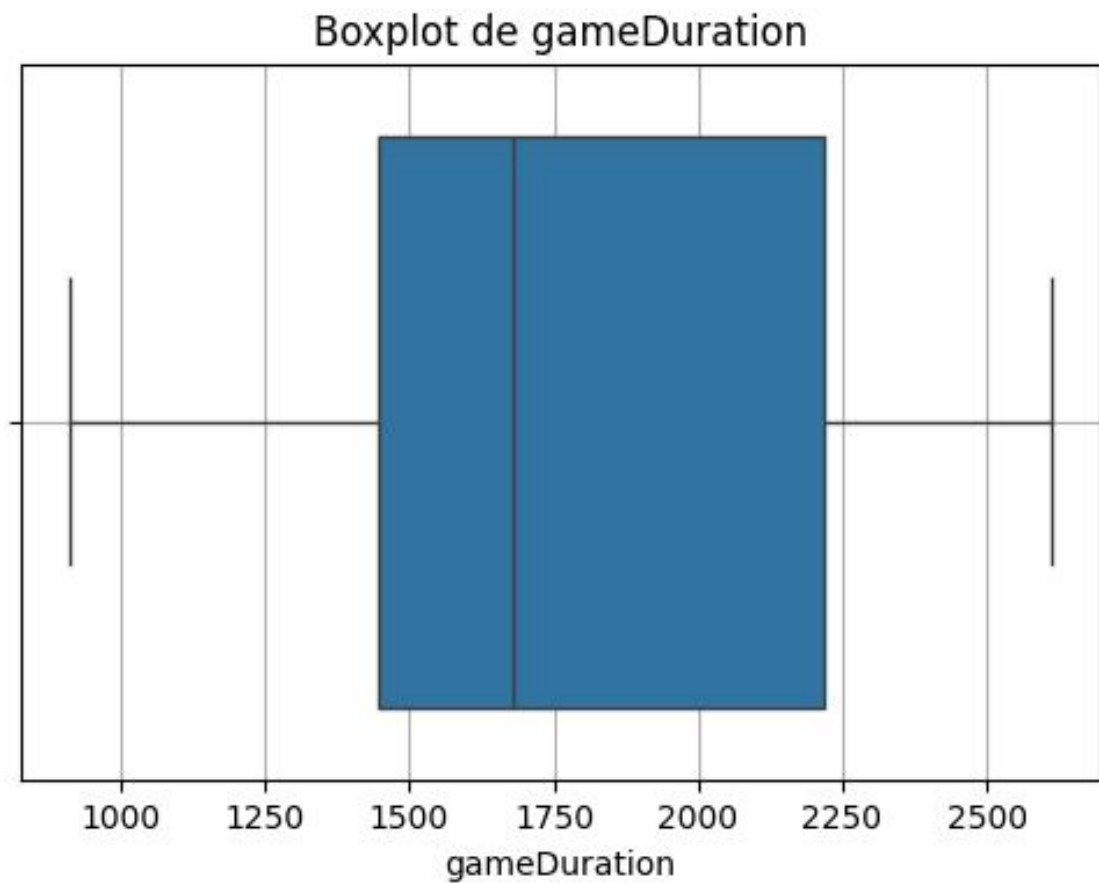
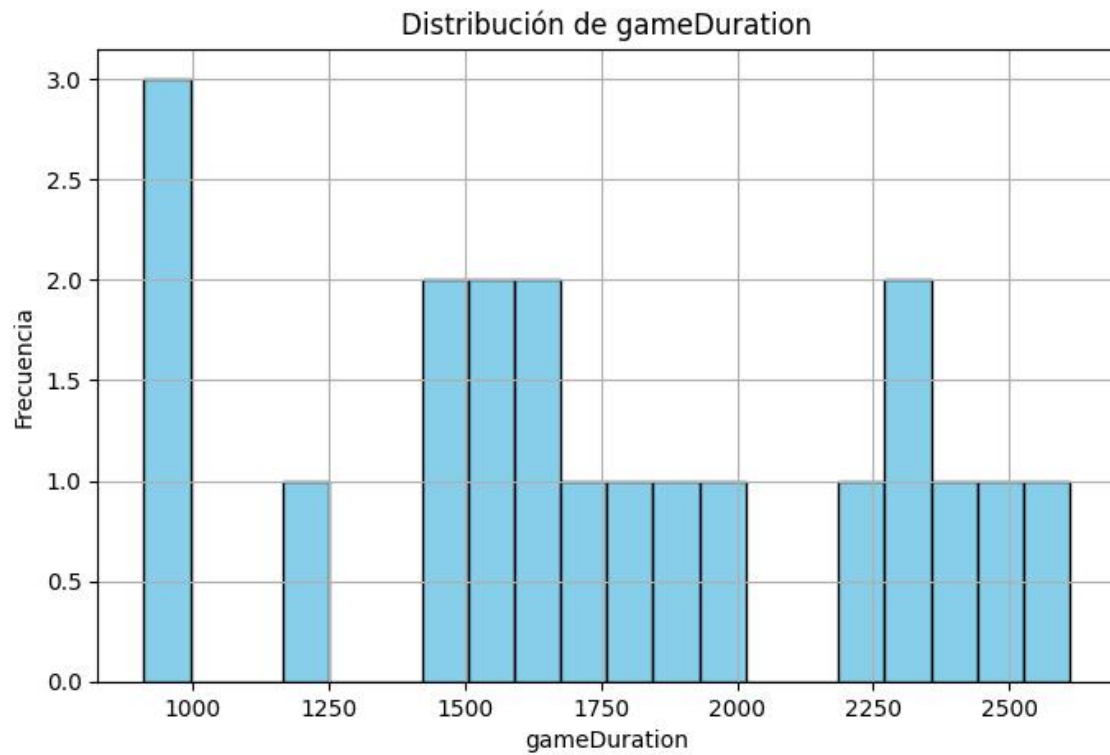




En promedio, rerzd obtiene 6.4 kills, muere 6.05 veces y logra 7.1 asistencias por partida.

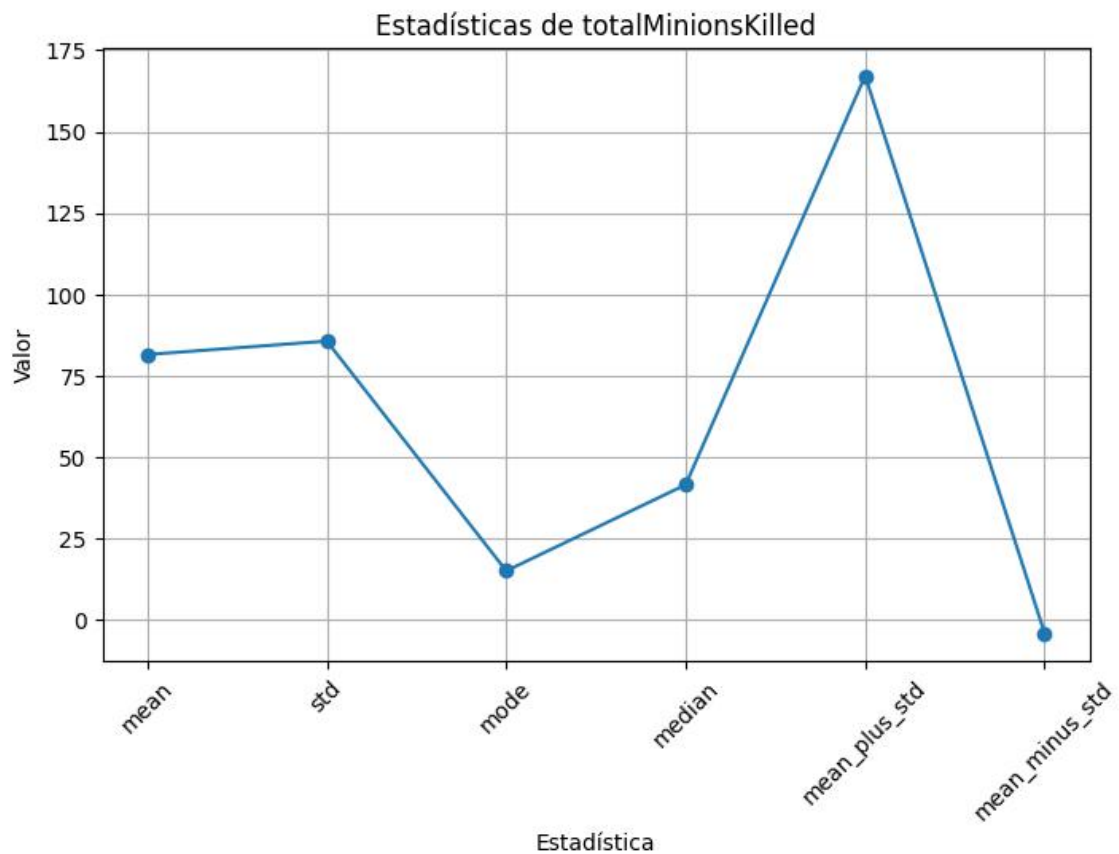
Sin embargo, hay una gran variabilidad (sd de 3.6 en kills y 5.02 en asistencias), indicando que en algunas partidas destaca mucho más que en otras.

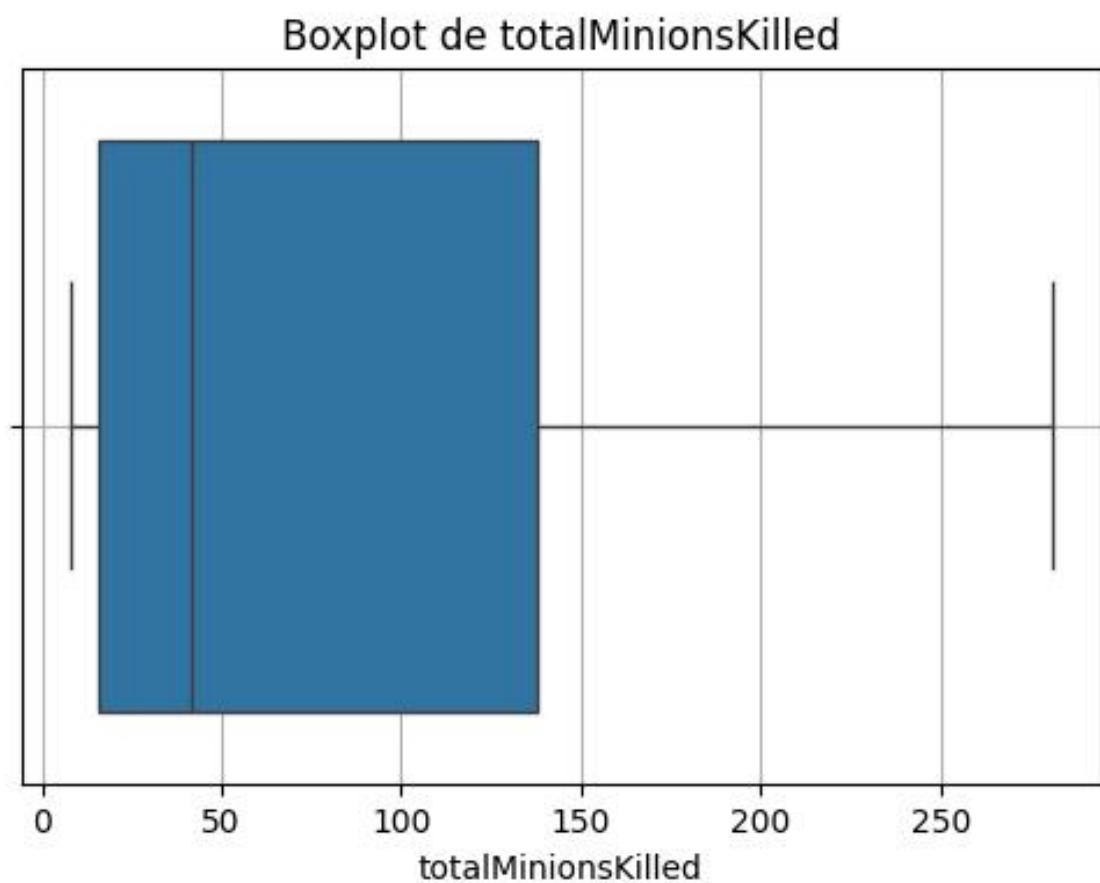
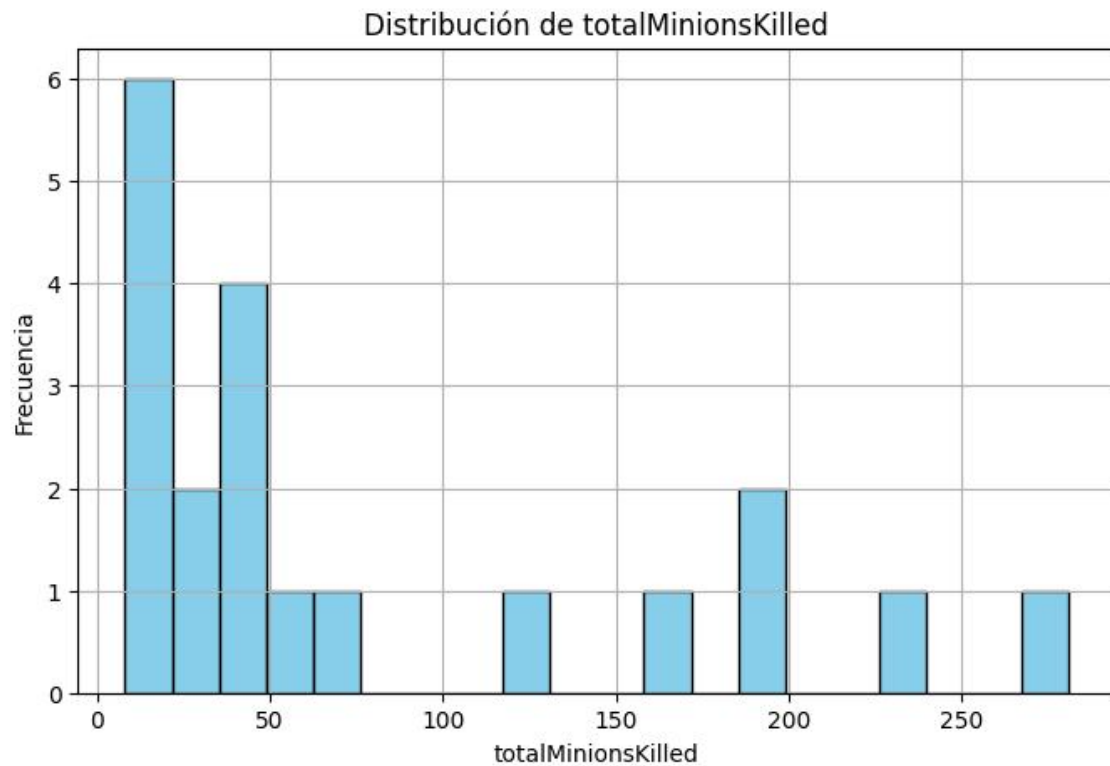




La duración promedio es de 29 minutos, con un rango entre 15 minutos y 43 min.

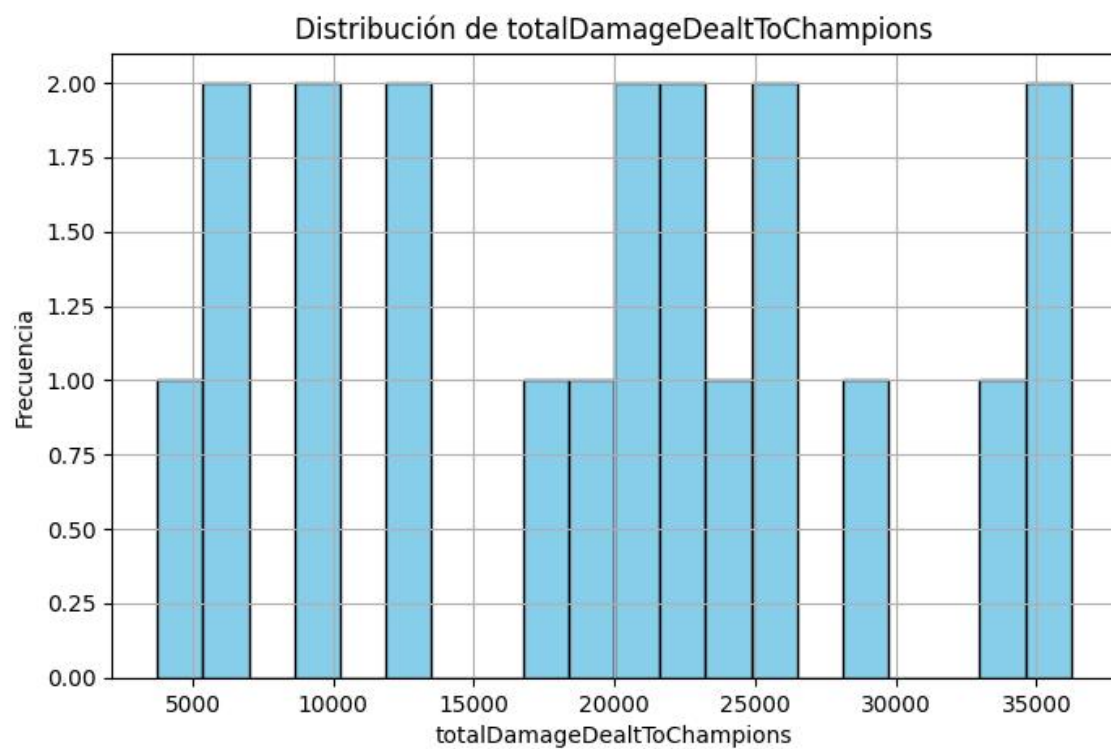
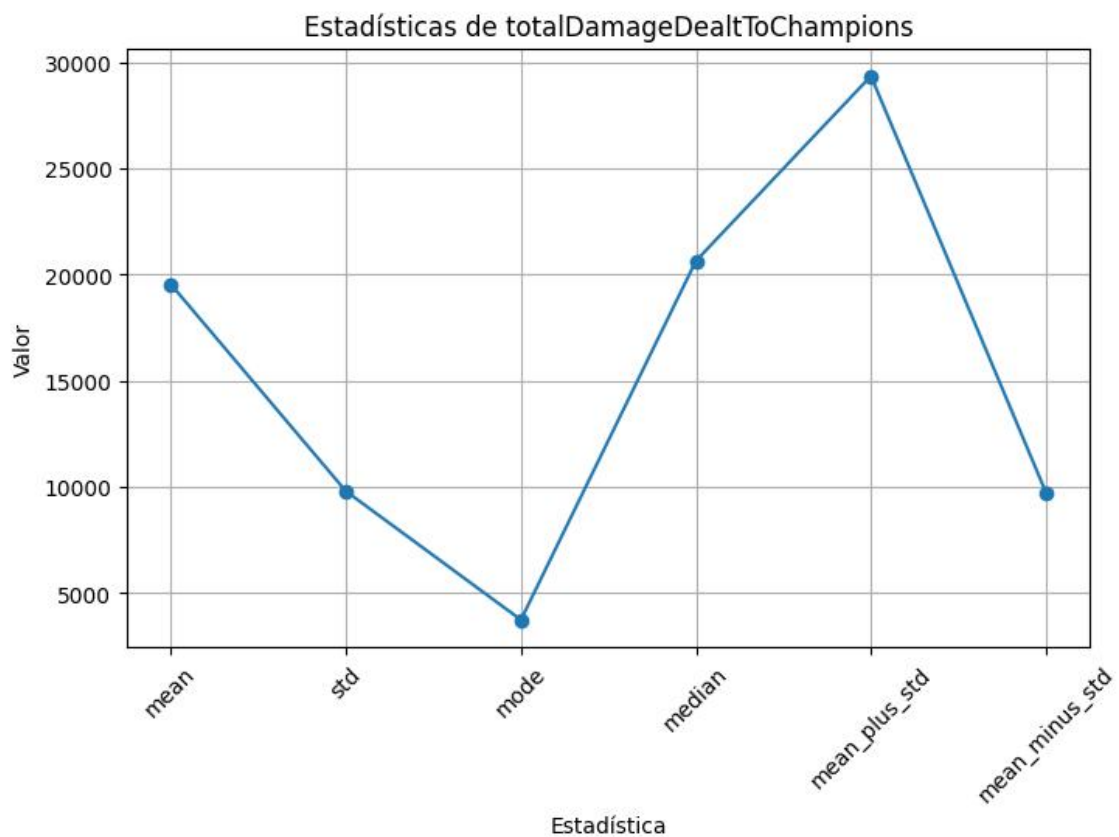
Hay una alta dispersión en los tiempos de juego (sd de aproximadamente 9 minutos), lo que sugiere que algunas partidas son rápidas mientras que otras se extienden bastante.

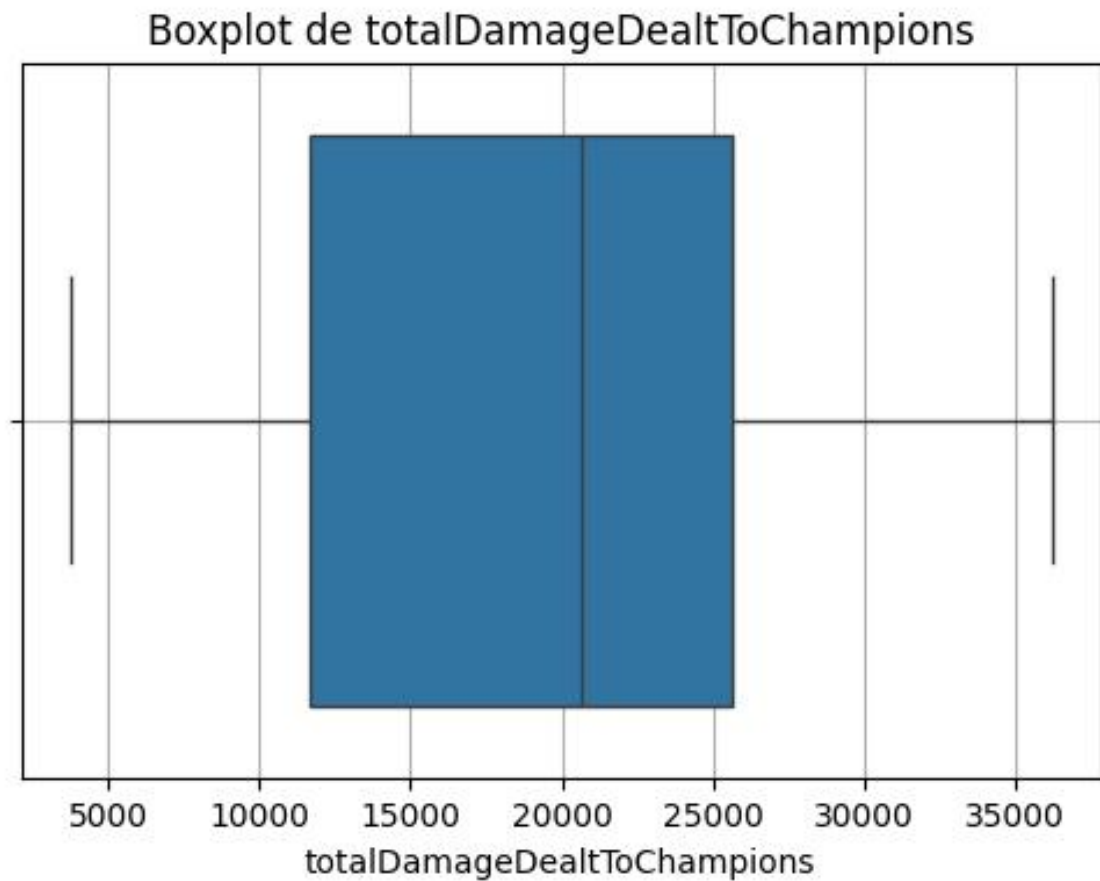




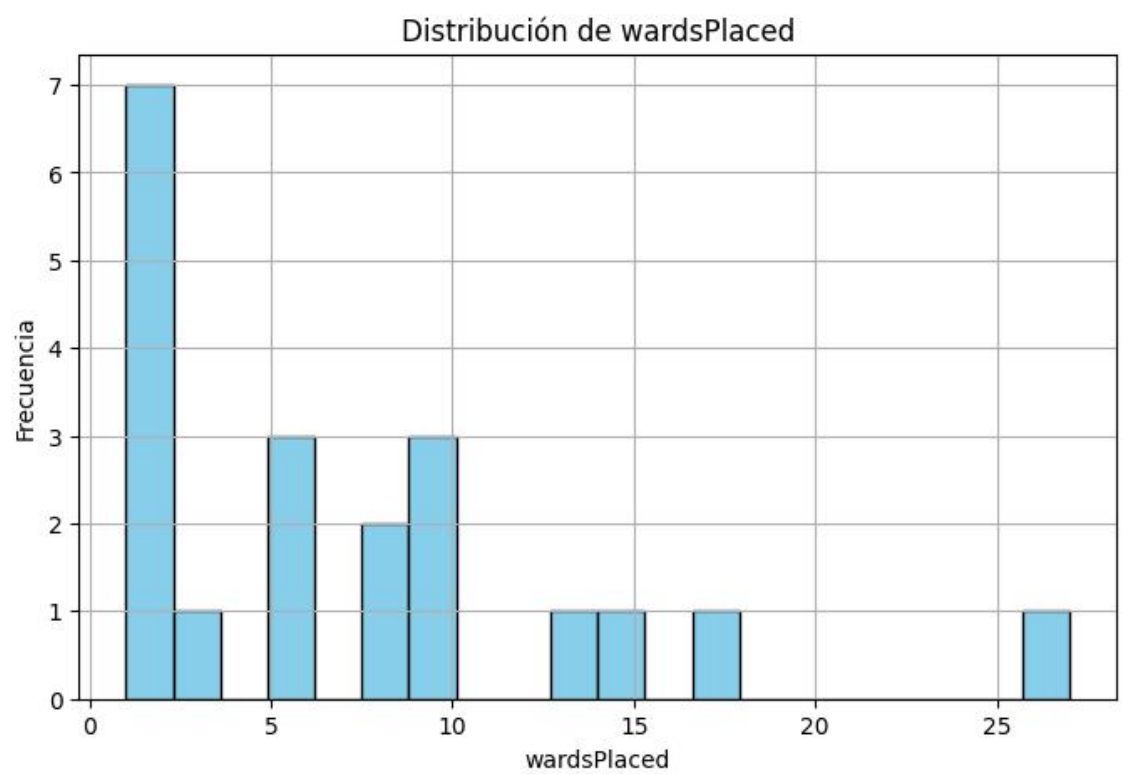
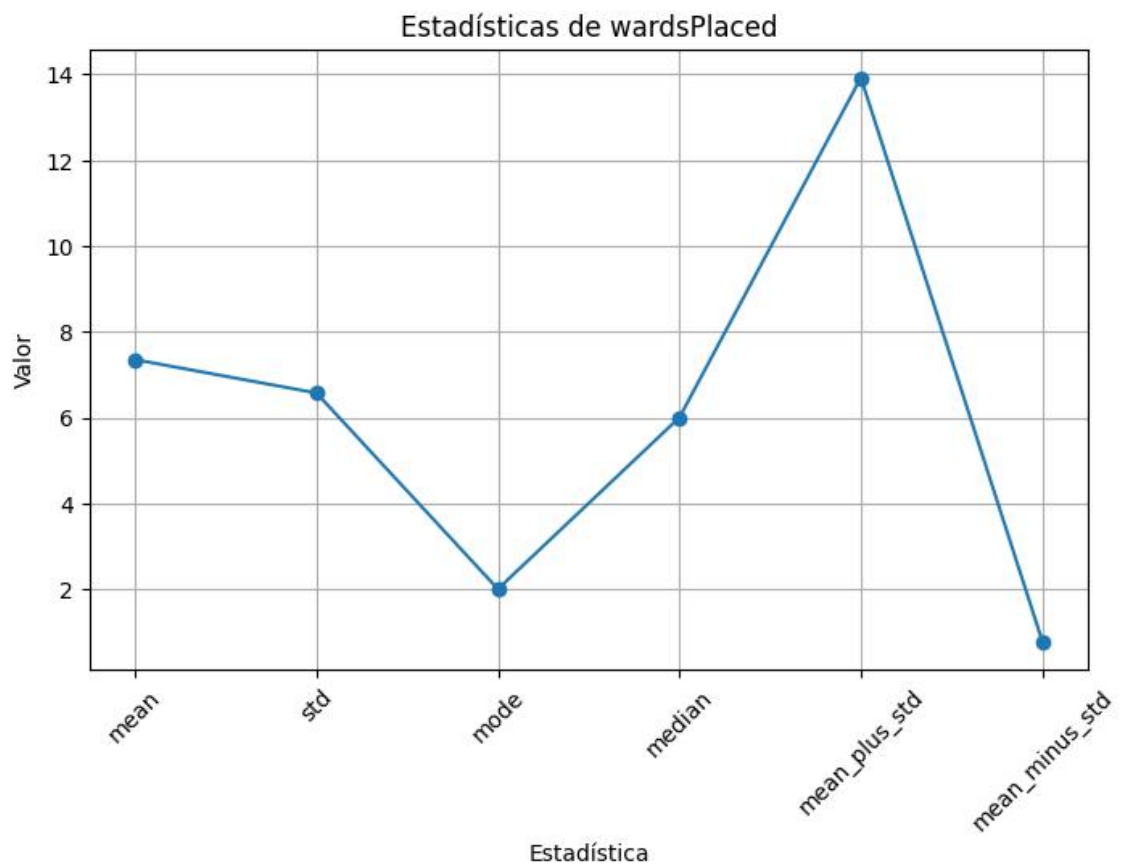
El promedio de súbditos eliminados es de 81.5, pero con una alta variabilidad (SD de 85.6), lo que indica

diferencias significativas entre las partidas en cuanto a su capacidad de farreo.

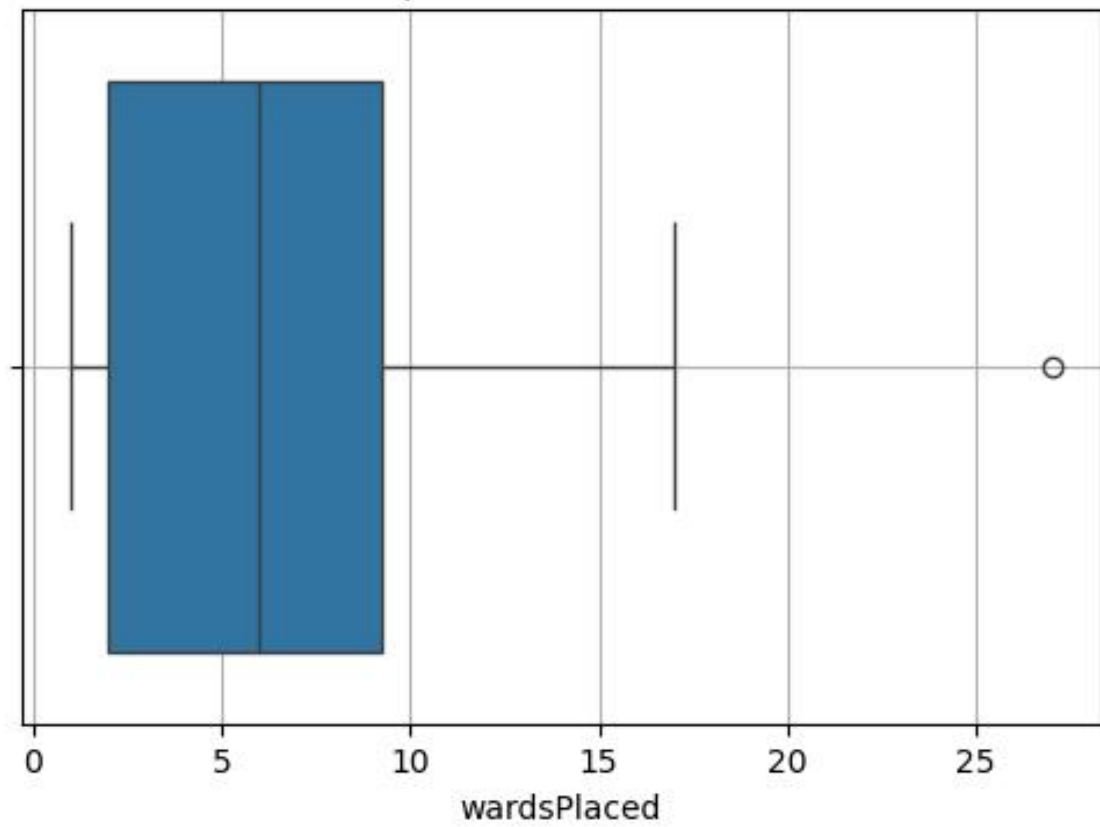




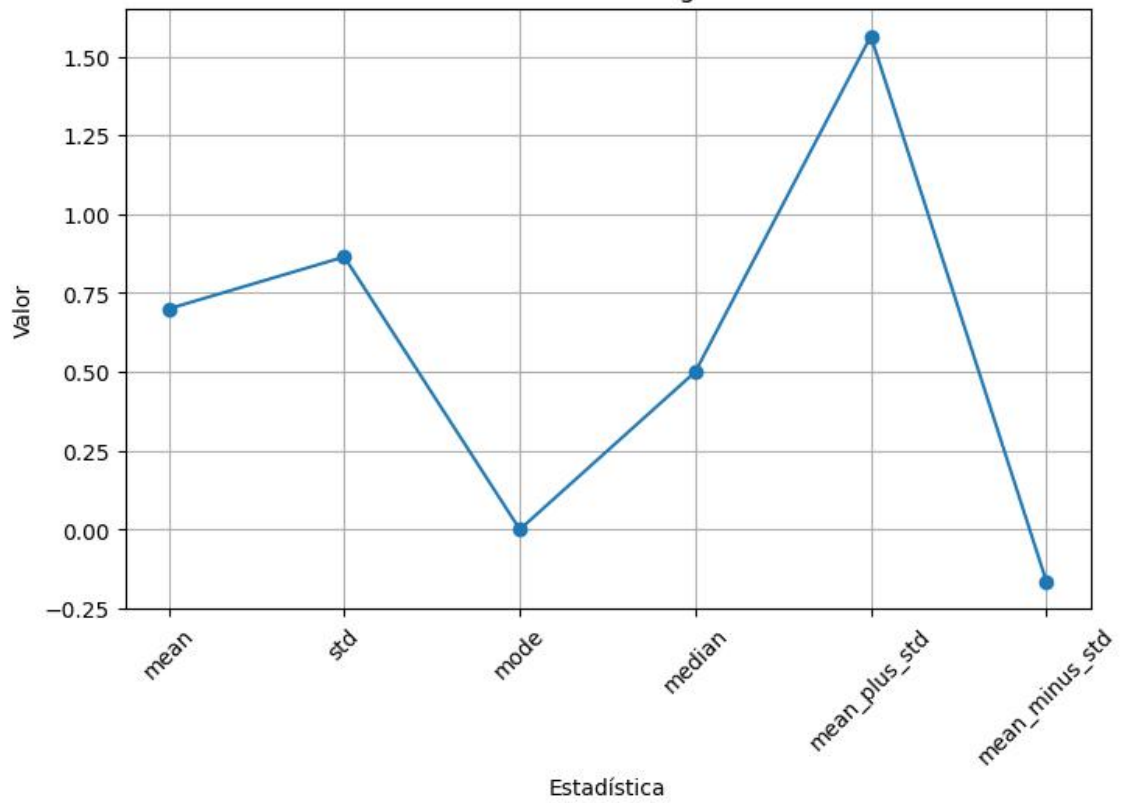
El daño promedio a campeones es de 19,544.8, con un mínimo de 3,747 y un máximo de 36,231, sugiriendo diferencias marcadas entre partidas en el impacto durante el combate.

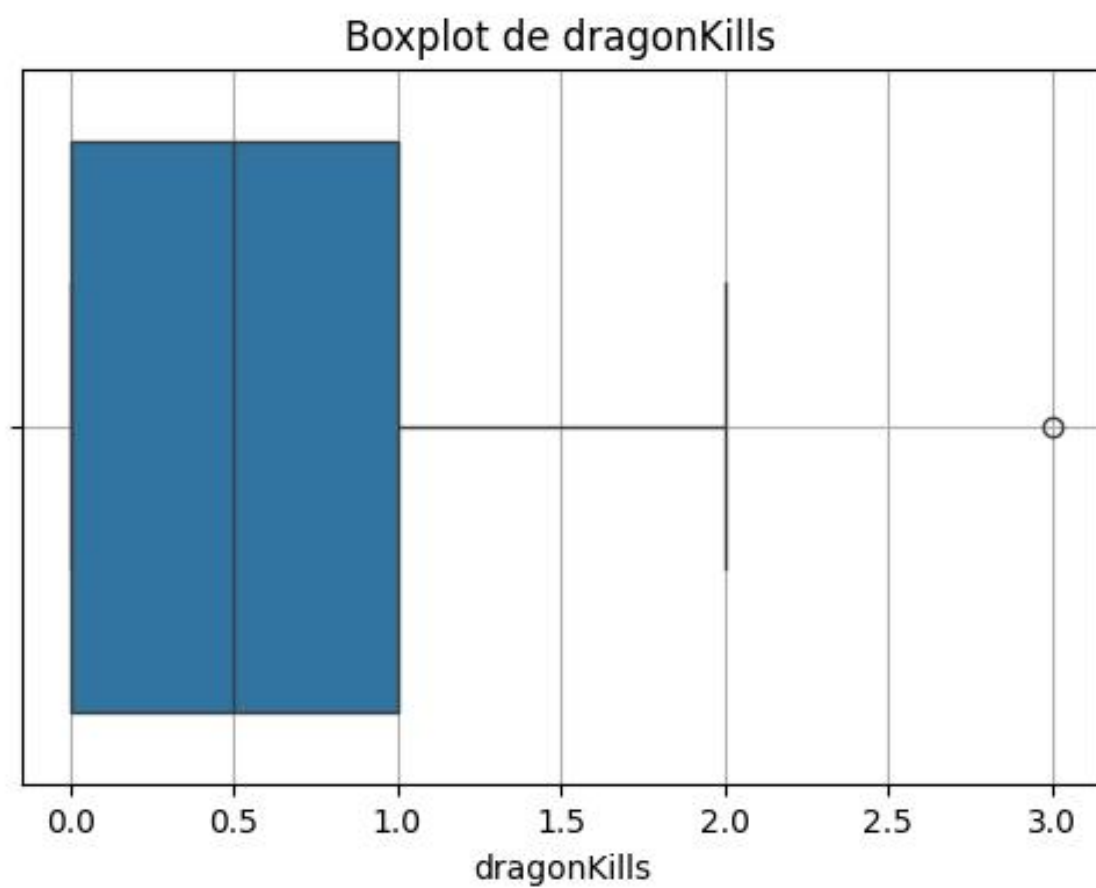
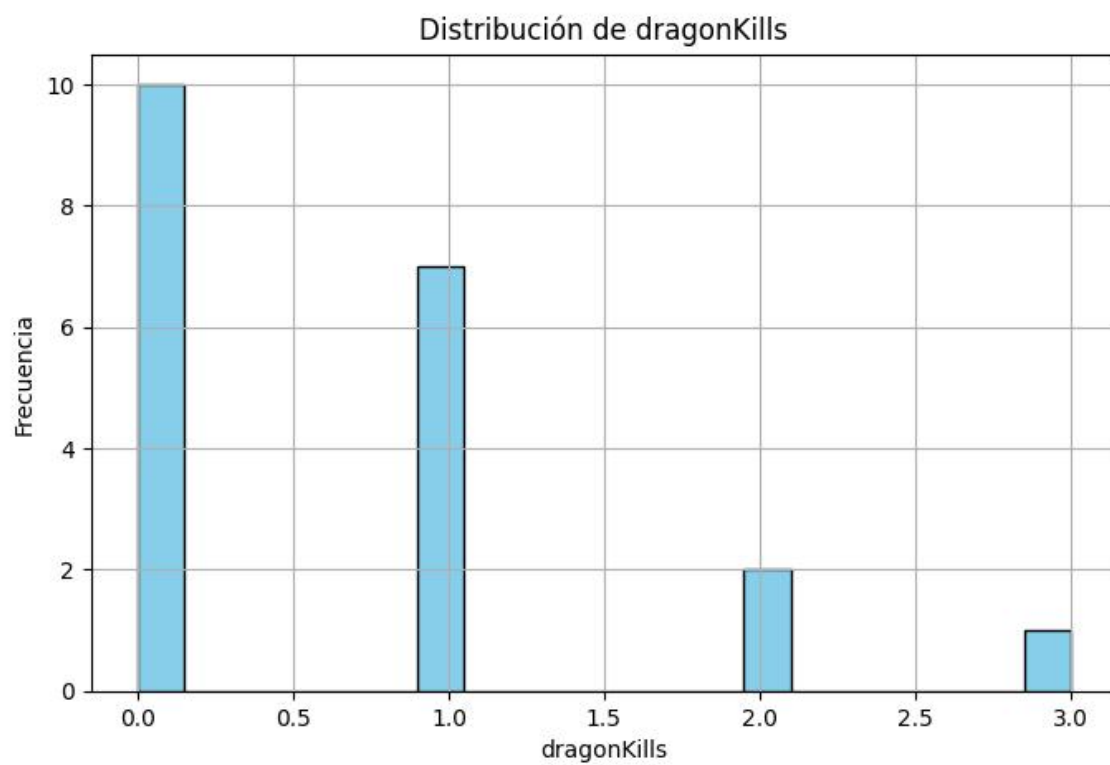


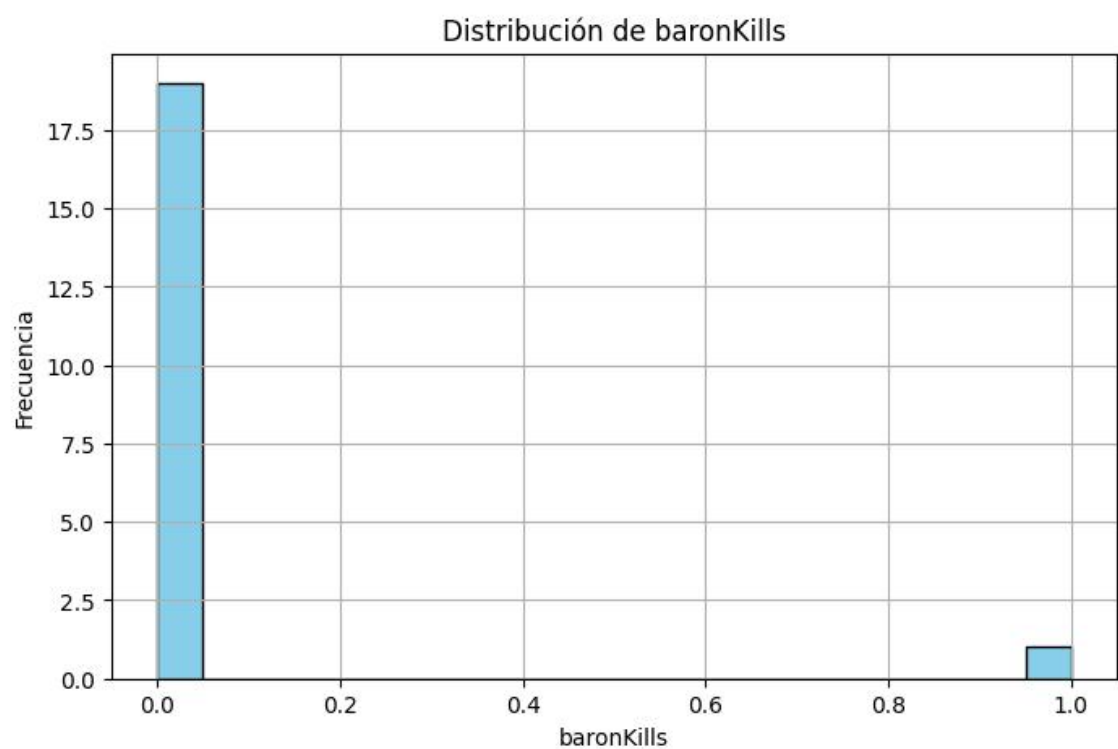
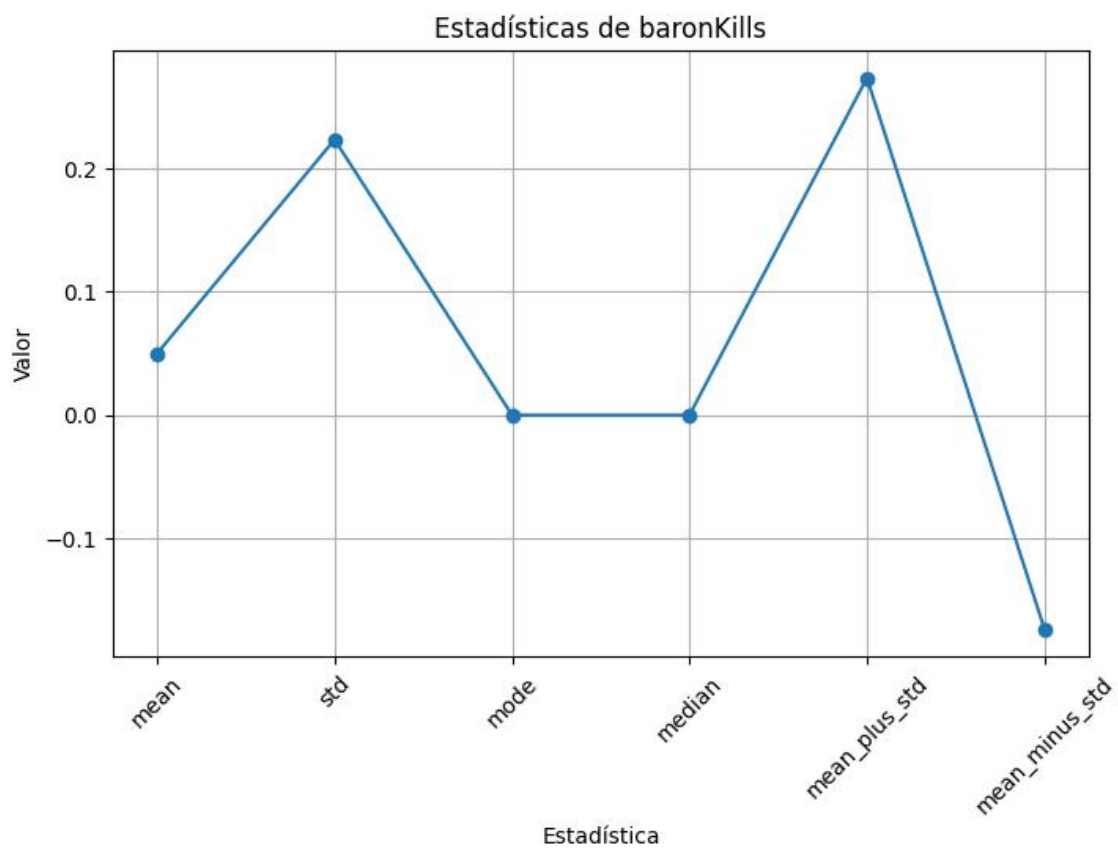
Boxplot de wardsPlaced

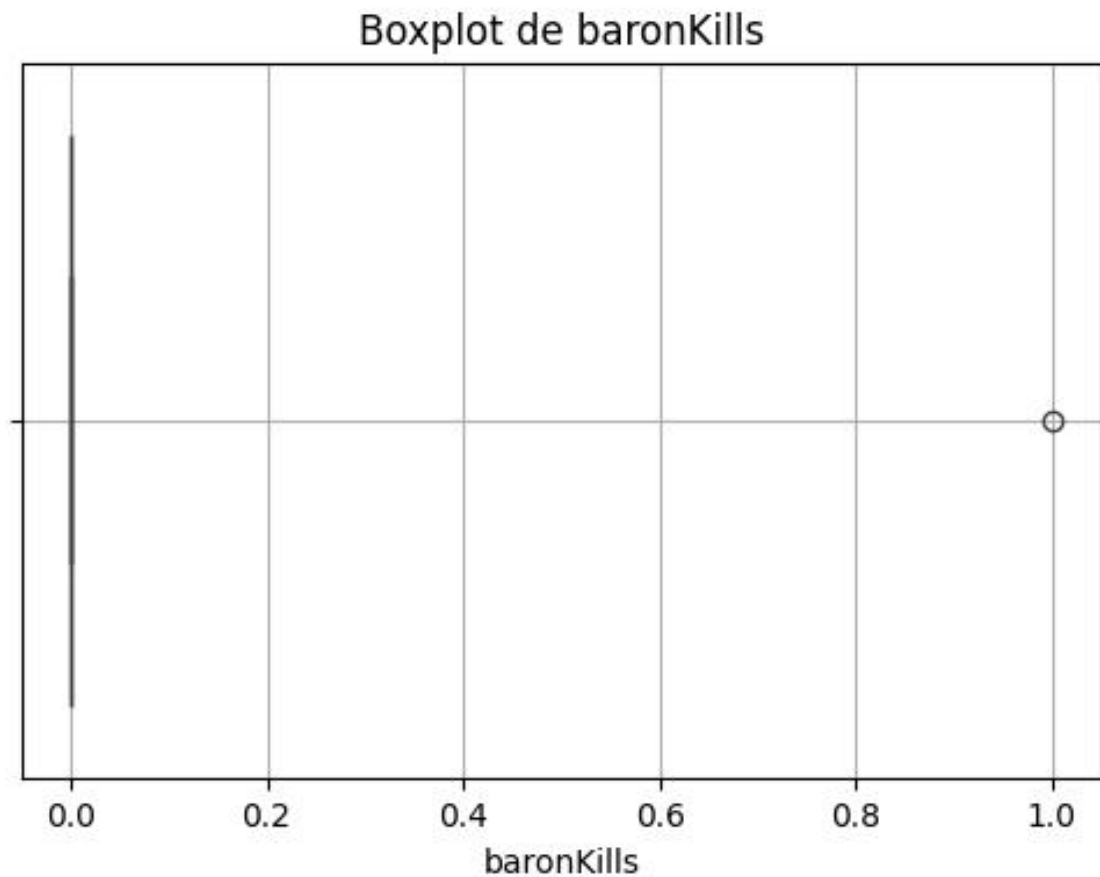


Estadísticas de dragonKills

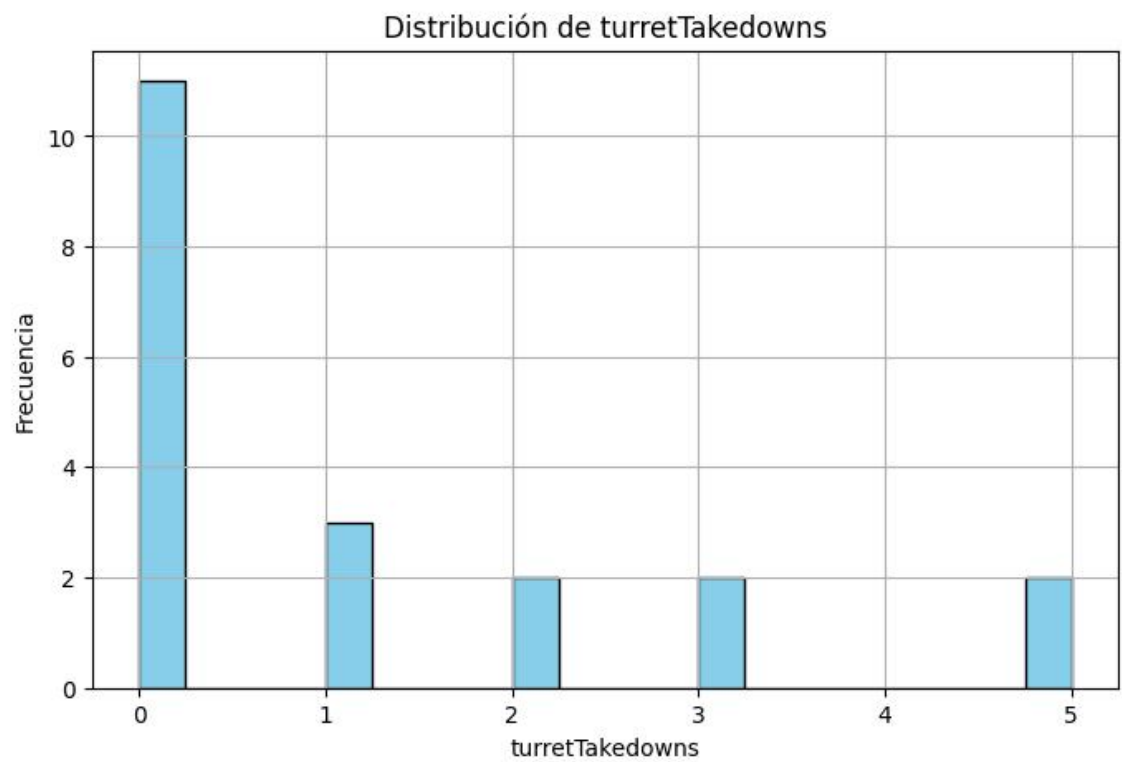
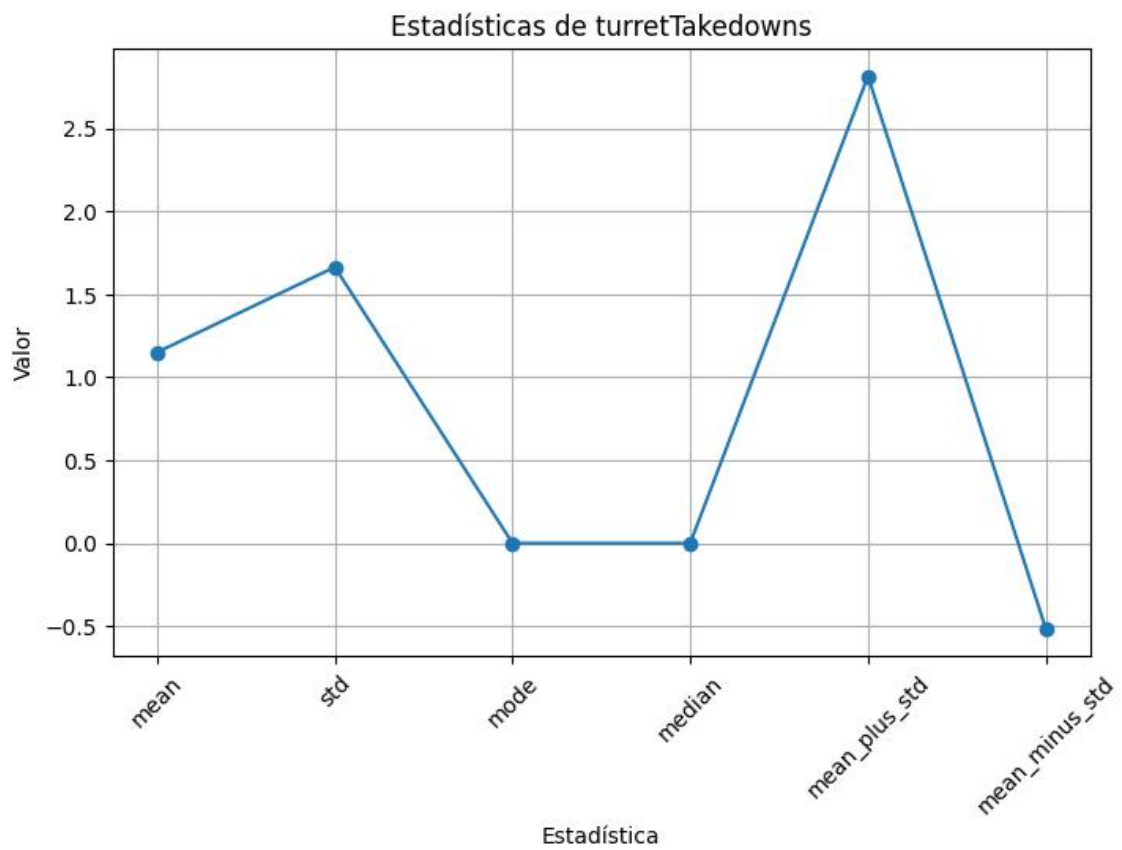


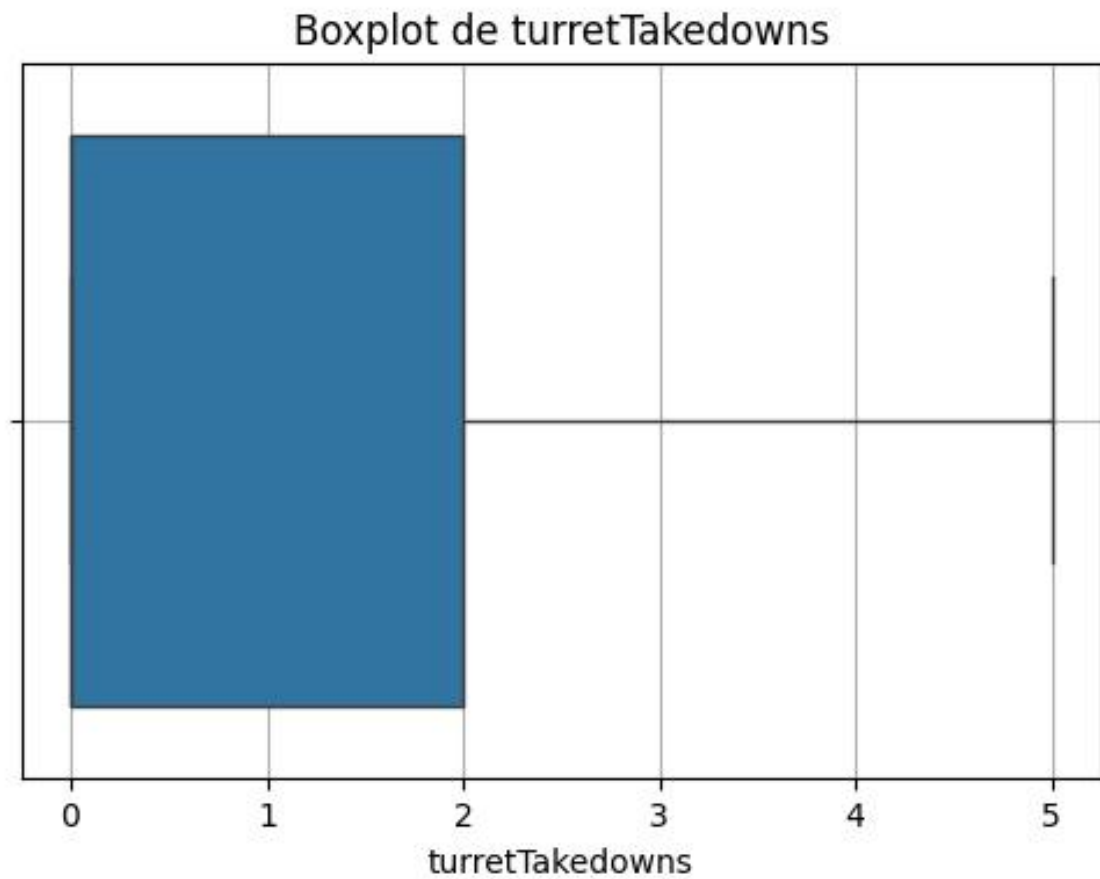




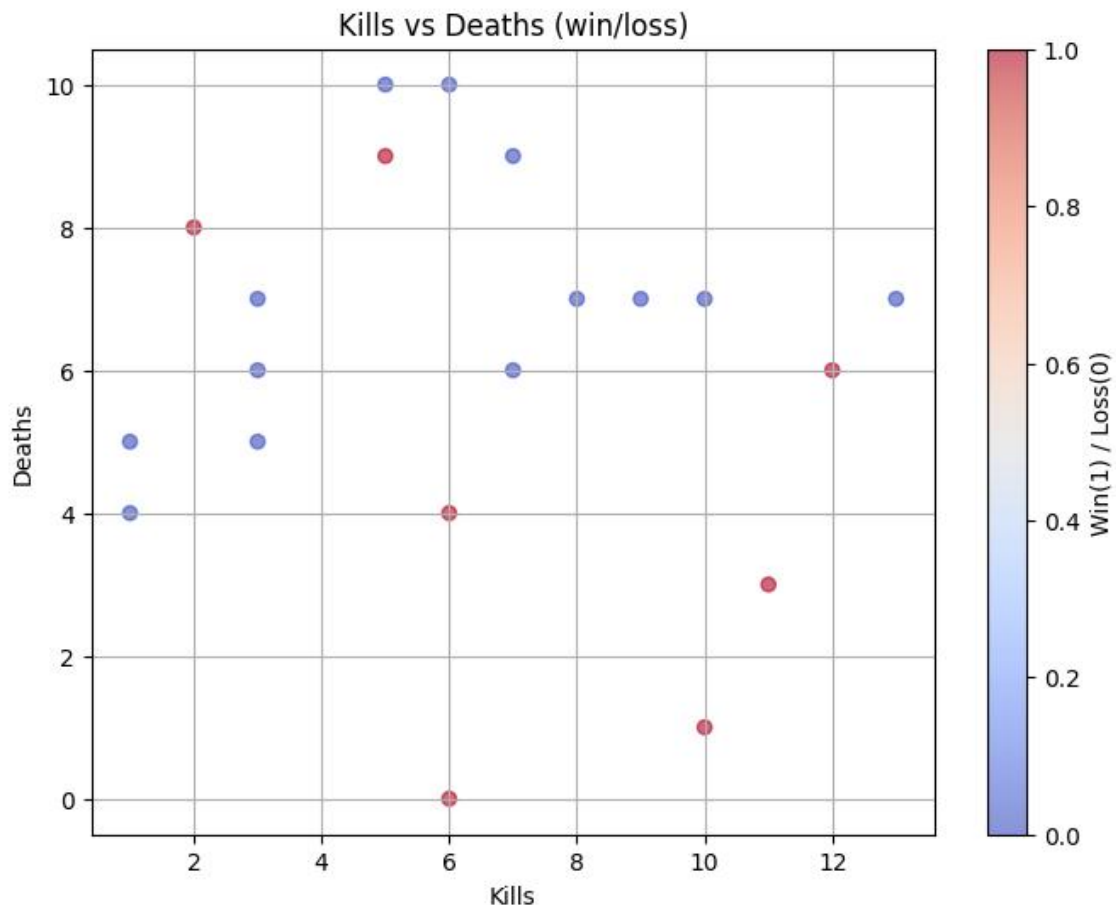


Las contribuciones a dragones y barones son bajas (0.7 y 0.05 en promedio, respectivamente), lo que sugiere que la mayoría de las partidas tienen una baja participación en estos objetivos.





La cantidad de torretas destruidas por rerzd es baja (1.15 en promedio) y la mayoría de las partidas no destruye torretas en sus partidas (mediana de 0).



Los puntos azules (derrotas) son más frecuentes en zonas donde hay muchas deaths (por ejemplo, más de 7 u 8), incluso si tienen kills decentes. Esto sugiere que morir demasiado impacta negativamente en el resultado.

Hay una gran dispersión de datos, lo que indica una variedad de estilos de juego y desempeños. Se observa una concentración de puntos entre 4 y 10 kills y 4 y 10 deaths, lo que parece ser el rango más común.

Tablas Que Se Crean A Partir De Los Datos Para La Creación De La Matriz De Confusión:

| | win | kills | deaths | assists \ |
|-----------------------------|-----------|-----------|-----------|-----------|
| win | 1.000000 | 0.213994 | -0.451399 | 0.391665 |
| kills | 0.213994 | 1.000000 | -0.088204 | 0.006367 |
| deaths | -0.451399 | -0.088204 | 1.000000 | 0.479857 |
| assists | 0.391665 | 0.006367 | 0.479857 | 1.000000 |
| gameDuration | 0.147956 | 0.395439 | 0.547202 | 0.654559 |
| totalMinionsKilled | 0.019468 | 0.364337 | -0.011932 | -0.093691 |
| totalDamageDealtToChampions | 0.055191 | 0.606387 | 0.348299 | 0.336115 |
| wardsPlaced | 0.189027 | -0.112431 | 0.049310 | 0.413257 |
| dragonKills | 0.385666 | -0.026918 | -0.195860 | 0.164765 |
| baronKills | -0.168345 | -0.026018 | 0.343800 | 0.416868 |
| turretTakedowns | 0.255449 | 0.269364 | 0.103568 | 0.111470 |
| firstBloodKill | -0.244600 | 0.056705 | 0.309835 | 0.333472 |
| firstTowerKill | 0.104828 | 0.198467 | -0.132786 | -0.074861 |

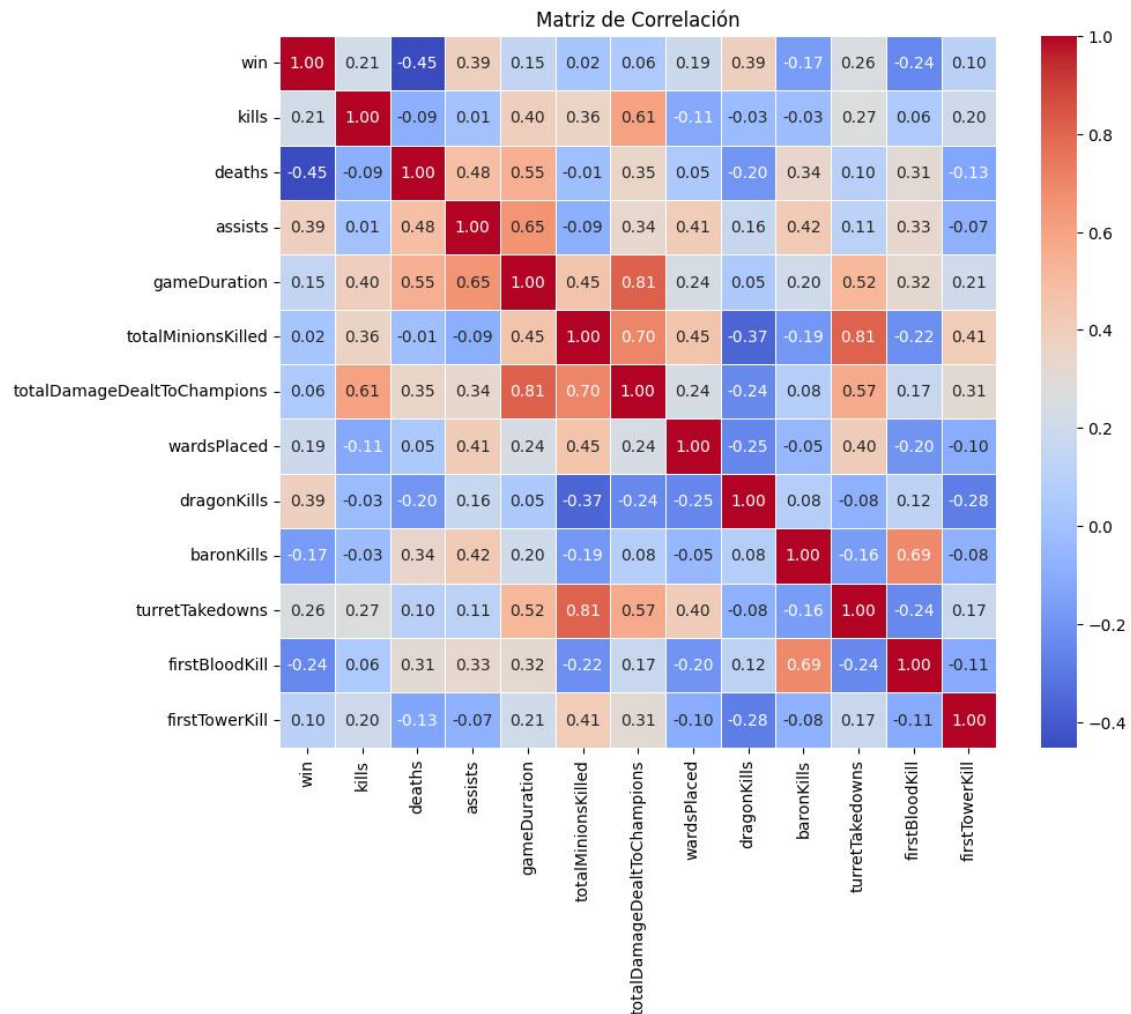
| | gameDuration | totalMinionsKilled \ |
|-----------------------------|--------------|----------------------|
| win | 0.147956 | 0.019468 |
| kills | 0.395439 | 0.364337 |
| deaths | 0.547202 | -0.011932 |
| assists | 0.654559 | -0.093691 |
| gameDuration | 1.000000 | 0.454929 |
| totalMinionsKilled | 0.454929 | 1.000000 |
| totalDamageDealtToChampions | 0.813618 | 0.702061 |
| wardsPlaced | 0.237132 | 0.451139 |
| dragonKills | 0.048312 | -0.366154 |
| baronKills | 0.200841 | -0.188291 |
| turretTakedowns | 0.515082 | 0.809946 |
| firstBloodKill | 0.322662 | -0.219663 |
| firstTowerKill | 0.205118 | 0.409373 |

| | totalDamageDealtToChampions | wardsPlaced \ |
|-----------------------------|-----------------------------|---------------|
| win | 0.055191 | 0.189027 |
| kills | 0.606387 | -0.112431 |
| deaths | 0.348299 | 0.049310 |
| assists | 0.336115 | 0.413257 |
| gameDuration | 0.813618 | 0.237132 |
| totalMinionsKilled | 0.702061 | 0.451139 |
| totalDamageDealtToChampions | 1.000000 | 0.236053 |
| wardsPlaced | 0.236053 | 1.000000 |
| dragonKills | -0.241431 | -0.249204 |
| baronKills | 0.084921 | -0.048353 |
| turretTakedowns | 0.572825 | 0.399464 |
| firstBloodKill | 0.173832 | -0.200356 |
| firstTowerKill | 0.313586 | -0.096275 |

| | dragonKills | baronKills | turretTakedowns | \ |
|-----------------------------|-------------|------------|-----------------|---|
| win | 0.385666 | -0.168345 | 0.255449 | |
| kills | -0.026918 | -0.026018 | 0.269364 | |
| deaths | -0.195860 | 0.343800 | 0.103568 | |
| assists | 0.164765 | 0.416868 | 0.111470 | |
| gameDuration | 0.048312 | 0.200841 | 0.515082 | |
| totalMinionsKilled | -0.366154 | -0.188291 | 0.809946 | |
| totalDamageDealtToChampions | -0.241431 | 0.084921 | 0.572825 | |
| wardsPlaced | -0.249204 | -0.048353 | 0.399464 | |
| dragonKills | 1.000000 | 0.081680 | -0.076876 | |
| baronKills | 0.081680 | 1.000000 | -0.162761 | |
| turretTakedowns | -0.076876 | -0.162761 | 1.000000 | |
| firstBloodKill | 0.118678 | 0.688247 | -0.236486 | |
| firstTowerKill | -0.276916 | -0.076472 | 0.174794 | |

| | firstBloodKill | firstTowerKill |
|-----------------------------|----------------|----------------|
| win | -0.244600 | 0.104828 |
| kills | 0.056705 | 0.198467 |
| deaths | 0.309835 | -0.132786 |
| assists | 0.333472 | -0.074861 |
| gameDuration | 0.322662 | 0.205118 |
| totalMinionsKilled | -0.219663 | 0.409373 |
| totalDamageDealtToChampions | 0.173832 | 0.313586 |
| wardsPlaced | -0.200356 | -0.096275 |
| dragonKills | 0.118678 | -0.276916 |
| baronKills | 0.688247 | -0.076472 |
| turretTakedowns | -0.236486 | 0.174794 |
| firstBloodKill | 1.000000 | -0.111111 |
| firstTowerKill | -0.111111 | 1.000000 |

Matriz De Correlación



Kills y totaldamagedealttochampions (0.606) tienen una fuerte relación, indicando que los jugadores con más daño infligen suelen conseguir más asesinatos.

Asistencias y wardsplaced (0.41) muestran que los jugadores que colocan más visión también tienden a ayudar más en equipo.

Gameduration y totalminionskilled (0.45) indican que partidas más largas suelen permitir una mayor acumulación de súbditos.

Conseguir la primera sangre no parece tener una gran influencia en la victoria (-0.24 con ganar), lo que sugiere que el desempeño general del equipo es más importante que un buen inicio.

Asegurar la primera torre tiene una correlación leve con la victoria (0.10 con ganar), lo que implica que es un buen indicador, pero no determinante.

Existe una correlación positiva entre dragónkills (0.385) y la probabilidad de ganar, lo que indica que asegurar dragones puede influir en la victoria.

Asistencias (0.39) también están correlacionadas positivamente con ganar, sugiriendo que jugar en equipo es clave.

Turret takedowns (0.25) tienen una relación moderada con la victoria, pero menos significativa que otros factores como dragones.

Curiosamente, baronkills (-0.168) tiene una correlación negativa con ganar en este conjunto de datos, lo que puede deberse a que los equipos que van perdiendo intentan forzar baron como una estrategia de recuperación

```
clf, accuracy = train_model(df)
```

```
save_model(clf, df, accuracy)
```

Resultado generado:
Model accuracy: 75.0

- ✓ Modelo guardado como
modelos_guardados/modelo_75.pkl
- ✓ Datos guardados como
modelos_guardados/datos_75.csv