columns\_to\_drop = {

'df\_game': ['team\_abbreviation\_home', 'team\_name\_home', 'video\_available\_away', 'game\_date', 'matchup\_home',

'matchup\_away', 'team\_abbreviation\_away', 'team\_name\_away', 'video\_available\_home'],

'df\_team': ['year\_founded'],

'df\_team\_details': ['id', "dleagueaffiliation"],

'df\_other\_stats': ['league\_id', 'team\_abbreviation\_home', 'team\_city\_home', 'team\_abbreviation\_away', 'team\_city\_away',

'largest\_lead\_home', 'largest\_lead\_away', 'lead\_changes', 'times\_tied', 'team\_turnovers\_home', 'team\_turnovers\_away',

'team\_rebounds\_home', 'team\_rebounds\_away'],

'df\_line\_score': ["pts\_ot5\_home","pts\_ot6\_home","pts\_ot7\_home","pts\_ot8\_home","pts\_ot9\_home","pts\_ot10\_home","pts\_ot5\_away", "pts\_ot6\_away","pts\_ot7\_away","pts\_ot8\_away","pts\_ot9\_away","pts\_ot10\_away", "game\_sequence","team\_abbreviation\_home", "team\_city\_name\_home","team\_nickname\_home","team\_abbreviation\_away","team\_city\_name\_away","team\_nickname\_away"],

'df\_common\_player': ['display\_first\_last', 'display\_last\_comma\_first', 'display\_fi\_last', 'player\_slug', 'last\_affiliation',

'team\_id', 'team\_code', 'dleague\_flag', 'nba\_flag', 'games\_played\_flag', 'greatest\_75\_flag',

'games\_played\_current\_season\_flag','school', 'team\_abbreviation', 'team\_city'],

'df\_draft\_history': ['player\_profile\_flag', 'draft\_type', 'player\_name', 'season', 'round\_number', 'round\_pick', 'overall\_pick',

'team\_city', 'team\_name', 'team\_abbreviation', 'organization', 'organization\_type'],

'df\_game\_info': ['game\_time'],

'df\_play\_by\_play': ['wctimestring', 'eventnum', 'neutraldescription', 'person1type', 'person2type', 'person3type',

'player1\_team\_city', 'player1\_team\_nickname', 'player2\_team\_city', 'player2\_team\_nickname', 'player3\_team\_city',

'player3\_team\_nickname', 'video\_available\_flag', 'player3\_id', 'player3\_name', 'player3\_team\_id',

'player3\_team\_abbreviation', 'eventnum', 'eventmsgactiontype', 'pctimestring', 'scoremargin', 'player1\_name',

'player1\_team\_id', 'player2\_name','player2\_team\_id']

#importar librerias

import pandas as pd

import numpy as np

#lecura de archivos a usar (conectar a la API, Seleccionar de un Bucket)

df\_game

df\_team

df\_team\_details

df\_other\_stats

df\_line\_score

df\_common\_player

df\_draft\_history

df\_game\_info

df\_play\_by\_play

#limpieza

dfs = dfs = {

'df\_game': df\_game,

'df\_team': df\_team,

'df\_team\_details': df\_team\_details,

'df\_other\_stats': df\_other\_stats,

'df\_line\_score': df\_line\_score,

'df\_common\_player': df\_common\_player,

'df\_draft\_history': df\_draft\_history,

'df\_game\_info': df\_game\_info,

'df\_play\_by\_play': df\_play\_by\_play

}

#eliminar columnas de dataframes

columns\_to\_drop = {

'df\_game': ['team\_abbreviation\_home', 'team\_name\_home', 'video\_available\_away', 'game\_date', 'matchup\_home',

'matchup\_away', 'team\_abbreviation\_away', 'team\_name\_away', 'video\_available\_home'],

'df\_team': ['year\_founded'],

'df\_team\_details': ['id', "dleagueaffiliation"],

'df\_other\_stats': ['league\_id', 'team\_abbreviation\_home', 'team\_city\_home', 'team\_abbreviation\_away', 'team\_city\_away',

'largest\_lead\_home', 'largest\_lead\_away', 'lead\_changes', 'times\_tied', 'team\_turnovers\_home', 'team\_turnovers\_away',

'team\_rebounds\_home', 'team\_rebounds\_away'],

'df\_line\_score': ["pts\_ot5\_home","pts\_ot6\_home","pts\_ot7\_home","pts\_ot8\_home","pts\_ot9\_home","pts\_ot10\_home","pts\_ot5\_away",

"pts\_ot6\_away","pts\_ot7\_away","pts\_ot8\_away","pts\_ot9\_away","pts\_ot10\_away", "game\_sequence","team\_abbreviation\_home",

"team\_city\_name\_home","team\_nickname\_home","team\_abbreviation\_away","team\_city\_name\_away","team\_nickname\_away"],

'df\_common\_player': ['display\_first\_last', 'display\_last\_comma\_first', 'display\_fi\_last', 'player\_slug', 'last\_affiliation',

'team\_id', 'team\_code', 'dleague\_flag', 'nba\_flag', 'games\_played\_flag', 'greatest\_75\_flag',

'games\_played\_current\_season\_flag','school', 'team\_abbreviation', 'team\_city'],

'df\_draft\_history': ['player\_profile\_flag', 'draft\_type', 'player\_name', 'season', 'round\_number', 'round\_pick', 'overall\_pick',

'team\_city', 'team\_name', 'team\_abbreviation', 'organization', 'organization\_type'],

'df\_game\_info': ['game\_time'],

'df\_play\_by\_play': ['wctimestring', 'eventnum', 'neutraldescription', 'person1type', 'person2type', 'person3type',

'player1\_team\_city', 'player1\_team\_nickname', 'player2\_team\_city', 'player2\_team\_nickname', 'player3\_team\_city',

'player3\_team\_nickname', 'video\_available\_flag', 'player3\_id', 'player3\_name', 'player3\_team\_id',

'player3\_team\_abbreviation', 'eventnum', 'eventmsgactiontype', 'pctimestring', 'scoremargin', 'player1\_name',

'player1\_team\_id', 'player2\_name','player2\_team\_id']

}

#combinar

df\_teams = cobinar tablas df\_team y df\_team\_details

#conversion de datos nulos

for name, df in dfs.items():

print(f"DataFrame: {name}")

print(df.isnull().sum()) # Cantidad de nulos por columna

print(f"Total de nulos: {df.isnull().sum().sum()}\n") # Total de nulos en el DataFrame

columnas\_a\_llenar = ['team\_turnovers\_home', 'total\_turnovers\_home', 'team\_rebounds\_home', 'pts\_off\_to\_home', 'team\_turnovers\_away', 'total\_turnovers\_away',

'team\_rebounds\_away', 'pts\_off\_to\_away']

for columna in columnas\_a\_llenar:

df[columna] = df[columna].fillna(0)

print(df)