```
import numpy as np
from sympy import diff, Abs, N
import math

def evaluar_funcion(F, variables, valores):
  subs_dict = {var: val for var, val in zip(variables, valores)}
  return N(F.subs(subs_dict))

def calc_error_absoluto(f, variables, X, dX):
  derivadas_parciales = []
  subs_dict = {var: val for var, val in zip(variables, X)}

  for i in range(len(variables)):
    derivadas_parciales.append(Abs(diff(f, variables[i])) * dX[i])

  coefs = list(map(lambda dp: N(dp.subs(subs_dict)), derivadas_parciales))

  return sum(coefs)

def calc_desviacion_estandar(f, variables, X, dX):
  derivadas_parciales = []
  subs_dict = {var: val for var, val in zip(variables, X)}
  for i in range(len(variables)):
    derivadas_parciales.append(Abs(diff(f, variables[i])) * dX[i])

  coefs = np.array(list(map(lambda dp: N(dp.subs(subs_dict)), derivadas_parciales)))
  squart_coefs = coefs**2
  suma = sum(squart_coefs)
  return math.sqrt(suma)

def calcular_medicion_indirecta(f, variables, datos):
  return np.array([evaluar_funcion(f, variables, datos[0]), calc_desviacion_estandar(f, variables,
datos[0], datos[1])])

def evaluar_funcion_en_lista(F, variables, vector):
  def evaluar_funcion_con_datos(datos):
    return evaluar_funcion(F, variables, datos[0])
  return np.array(list(map(evaluar_funcion_con_datos, vector)))

def calcular_medicion_indirecta_en_lista(F, variables, vector):
  def calcular_medicion_indirecta_con_datos(datos):
    return calcular_medicion_indirecta(F, variables, datos)
  return np.array(list(map(calcular_medicion_indirecta_con_datos, vector)))

def separar_valores_incertidumbres(datos):
  valores = datos[::2]
  incertidumbres = datos[1::2]
  return np.array([valores, incertidumbres])

def separar_valores_incertidumbres_en_lista(vector):
  return np.array(list(map(separar_valores_incertidumbres, vector)))

def seleccionar_columnas_de_tabla(tabla, columnas):
  return tabla[:, columnas]


def agregar_calculo_a_dataframe(df, f, symbols, cols, nombreColumna):
  variables = symbols
  datos = df.to_numpy()
  datosRelevantes = seleccionar_columnas_de_tabla(datos, cols)
  datosRelevantesSeparados = separar_valores_incertidumbres_en_lista(datosRelevantes)
  resultado = calcular_medicion_indirecta_en_lista(f, variables, datosRelevantesSeparados)
  df[nombreColumna] = resultado[:,0]
  df['Î”' + nombreColumna] = resultado[:,1]
```