

Apéndice C: Código Matlab Simulación De La Planta De Biocarbon

```

%% =====
% MODELO PRINCIPAL DE SIMULACION DE LA PLANTA DE BIOCARON (15 AÑOS)
% BALANCE DE MASAS - VERSION LIMPIA
% =====

clear; clc;

%% -----
% 1. PARAMETROS GENERALES
% -----

n_anios      = 15;
semanas_por_anio = 52;
dias_por_anio  = 365;
dias_totales   = n_anios * dias_por_anio;

% Guardar en param para funciones externas
param.n_anios      = n_anios;
param.semanas_por_anio = semanas_por_anio;
param.dias_por_anio  = dias_por_anio;
param.dias_total     = dias_totales;

% Crecimiento anual de la produccion (factor acumulado por año):
vec_factor = zeros(n_anios,1);
vec_factor(1) = 1.0;
for a = 2:n_anios
    if a <= 9
        vec_factor(a) = vec_factor(a-1) * 1.1;    % +10 % años 2-9
    else
        vec_factor(a) = vec_factor(a-1) * 1.01;    % +1 % años 10-15
    end
end

% Recepcion base:
recepcion_base_ton_dia = 4;    % [ton/dia] por jornada de recoleccion
recepcion_base_sem_ton = 2 * recepcion_base_ton_dia; % [ton/sem] (lunes+viernes)

```

```

% ----- Parametros especificos de equipos (solo MASA / CAPACIDAD) -----

% Tolva (A2)
fraccion_escurrimiento_tolva = 0.02;    % 2% del peso se va como liquido (L)

% Banda de seleccion manual (A3)
fraccion_rechazo_banda = 0.01;          % 1% del peso se rechaza
vel_banda_kg_h          = 1333.3;        % [kg/h] ~ 4 ton / 3 h (para energia luego)

% Molino triturador (A4)
vel_molino_kg_h          = 1333.3;        % [kg/h] (mismo ritmo de linea, sin
perdidas de masa)

% Tornillo sin fin (A5)
vel_tornillo_kg_h        = 1333.3;        % [kg/h] (mismo ritmo, sin perdidas de
masa)

% Pulverizador de biomasa (A8)
vel_pulverizador_kg_h    = 500;          % [kg/h] dato de diseño

% Zaranda / tromel giratorio (A9)
vel_zaranda_kg_h         = 500;          % [kg/h] asumida igual al pulverizador

% ===== CABAS (A6) =====

numero_cabas             = 6;
dias_retencion_cabas     = 3;            % [dias] de retencion en cabas
volumen_caba_m3           = 3.1;         % [m3] volumen util
densidad_biomasa_molino_kg_m3 = 936;     % [kg/m3] despues del molino

masa_max_caba_kg = densidad_biomasa_molino_kg_m3 * volumen_caba_m3; % ~2900 kg

% Datos de experimento de cabas (3 dias):
masa_inicial_exp = 14.96;    % [kg]
masa_final_exp   = 12.63;    % [kg]

```

```

vol_lix_exp_L      = 0.150;    % [L] lixiviado acumulado

% Fraccion de perdida total de masa en cabas en 3 dias (~15,6 %)
fraccion_perdida_caba = (masa_inicial_exp - masa_final_exp)/masa_inicial_exp;

% Lixiviado generado por kg de biomasa que entra a cabas [L/kg]
lixiviado_por_kg_caba_lkg = vol_lix_exp_L / 13.48 ;

%% ===== PARAMETROS DEL PATIO (A7) =====

largo_patio_m      = 16;
ancho_patio_m       = 8;
altura_capa_m       = 0.2;

volumen_patio_m3 = largo_patio_m * ancho_patio_m * altura_capa_m;

densidad_biomasa_patio_kg_m3 = 450;

masa_max_patio_kg = volumen_patio_m3 * densidad_biomasa_patio_kg_m3;

dias_retencion_patio = 3;

% Perdida en patio (compacto, sacado del experimento de secado solar)
fraccion_perdida_patio = (0.5115 - fraccion_perdida_caba);

% Humedad final del material seco
humedad_final_patio = 0.1034;

% ===== HORNO DE PIROLISIS (A10) =====

T_horno_C          = 250;    % [°C] temperatura de proceso (caso base)
capacidad_horno_kg_h = 500;    % [kg/h] capacidad nominal asumida

% Tabla experimental de rendimientos de pirolisis
T_exp = [ 0   175   200   225   250   275   300   350 ];

```

```

R_solid_pct = [100 87.6 77.5 70.3 59.6 55.8 52.9 41.3];
R_liquid_pct = [ 0 11.4 15.7 16.6 29.8 32.2 34.1 42.0];
R_gas_pct = [ 0 1.0 6.8 13.1 10.6 12.0 13.0 16.7];

% Convertimos a fraccion
R_solid_fr = R_solid_pct / 100;
R_liquid_fr = R_liquid_pct / 100;
R_gas_fr = R_gas_pct / 100;

% ===== A11: CARGADOR FRONTAL AGRICOLA =====
% Horas base de operacion en dias de maxima carga (caso base 4 t/dia)
H_carg_max_base_h_dia = 3.0; % [h/dia] para ~4 ton/dia

% Horas en dias de minima operacion (solo volteo / ajustes)
H_carg_min_h_dia = 1.0; % [h/dia]

% Tope maximo razonable de horas por dia
H_carg_max_tope_h_dia = 6.0; % [h/dia]

% Masa de referencia para dias de maxima operacion
masa_ref_intensa_kg_dia = 4000; % [kg/dia] referencia: 4 ton/dia

%% -----
% PARAMETROS MODELO_ENERGETICO_PLANTA
% -----

param.P_A3_kW = 2.2; param.eta_A3 = 0.875;
param.P_A4_kW = 15.0; param.eta_A4 = 0.882;
param.P_A5_kW = 2.2; param.eta_A5 = 0.875;
param.P_A8_kW = 4.0; param.eta_A8 = 0.90;
param.P_A9_kW = 2.2; param.eta_A9 = 0.875;

param.P_soplador_horno_kW = 0.25; param.eta_soplador = 0.90;
param.P_aux_horno_kW = 0.25; param.eta_aux_horno = 0.90;

param.P_A12_aux_kW = 0.75; param.eta_A12_aux = 0.90;
param.P_admin_kW = 0.50; param.eta_admin = 1.0;

param.PCI_GN_kWh_m3 = 10.0; % Representa la energia contenida en 1 m3 de gas
natural.

```

```

param.PCI_diesel_kWh_L = 9.9;    % Representa la energia contenida en 1 litro de
diesel.
param.consumo_diesel_L_h = 3.0;

% ----- QUEMADOR GN (modelo termodinámico 0D) -----

param.T_in_C    = 25;           % temperatura biomasa entrada [°C]
param.T_proc_C  = T_horno_C;    % temperatura de proceso [°C]
param.T_amb_C   = 25;           % temperatura ambiente [°C]
param.T_fg_C    = 350;          % gases de escape [°C]

% Fracción de pérdidas térmicas del horno
param.frac_perdidas = 0.15;      % 15 %

% Exceso de aire en combustión GN
param.lambda_air = 1.30;         % 30 %

% Propiedades térmicas biomasa / agua
param.cp_bio_J_kgK    = 1400;
param.cp_w_liq_J_kgK  = 4180;
param.cp_w_vap_J_kgK  = 2000;
param.h_vap_J_kg      = 2257000;

% Calor adicional por torrefacción (literatura)
param.Q_torref_J_kg_dry = 350000;

% ---- Gases de combustión (valores medios ~350°C) (623 K) ----
param.cp_CO2_kJ_kmolK = 40;
param.cp_H2O_kJ_kmolK = 36;
param.cp_O2_kJ_kmolK  = 31;
param.cp_N2_kJ_kmolK  = 31;

param.M_CO2 = 44;
param.M_H2O = 18;
param.M_O2  = 32;
param.M_N2  = 28;

%% -----
% 2. VECTORES PARA GUARDAR RESULTADOS ANUALES (RECEPCION)
% -----

```

```

vec_anio      = (1:n_anios)';      % columna de años
vec_Lun_ton   = zeros(n_anios,1); % toneladas por dia lunes
vec_Mie_ton   = zeros(n_anios,1); % toneladas por dia miercoles
vec_Vie_ton   = zeros(n_anios,1); % toneladas por dia viernes
vec_ton_sem   = zeros(n_anios,1); % toneladas por semana
vec_ton_anio  = zeros(n_anios,1); % toneladas por año
vec_Mar_ton   = zeros(n_anios,1); % martes
vec_Jue_ton   = zeros(n_anios,1); % jueves

%% -----
% 3. CALCULO DE RECEPCION SEMANAL POR AÑO
% -----

for anio = 1:n_anios
    factor = vec_factor(anio);

    % Recepcion semanal base
    ton_sem_obj = recepcion_base_sem_ton * factor;

% % ---- CONFIGURACION POR DIA ----
%     Lun_ton = 0; Mar_ton = 0; Mie_ton = 0; Jue_ton = 0; Vie_ton = 0;
%
%     % Opcion A igualar todos los dias activos :
%     % Lun-Vie constantes:
%     Lun_ton = ton_sem_obj/5; Mar_ton = ton_sem_obj/5;...
%     Mie_ton = ton_sem_obj/5; Jue_ton = ton_sem_obj/5; ...
%     Vie_ton = ton_sem_obj/5;

%Opcion B Lunes y viernes fijo , crecimiento el miercoles

    if ton_sem_obj <= 8
        Lun_ton = ton_sem_obj / 2;
        Mie_ton = 0;
        Vie_ton = ton_sem_obj / 2;

    elseif ton_sem_obj <= 12
        Lun_ton = 4;
        Vie_ton = 4;
        Mie_ton = ton_sem_obj - 8;

```

```

else
    Lun_ton = ton_sem_obj / 3;
    Mie_ton = ton_sem_obj / 3;
    Vie_ton = ton_sem_obj / 3;
end

vec_Lun_ton(anio) = Lun_ton;
vec_Mie_ton(anio) = Mie_ton;
vec_Vie_ton(anio) = Vie_ton;

    % Activar para la Opcion A, comentar para la B
    % vec_Mar_ton(anio) = Mar_ton;
    % vec_Jue_ton(anio) = Jue_ton;
    % vec_ton_sem(anio) = Lun_ton + Mie_ton + Vie_ton + Mar_ton + Jue_ton;

vec_ton_sem(anio) = Lun_ton + Mie_ton + Vie_ton;

vec_ton_anio(anio) = vec_ton_sem(anio) * semanas_por_anio;
end

%% -----
% 4. TABLA DE RESUMEN: RECEPCION SEMANAL Y ANUAL
% -----

tabla_recepcion = table( ...
    vec_anio, ...
    vec_factor, ...
    vec_Lun_ton, ...
    vec_Mar_ton, ... % comentar Opcion B
    vec_Mie_ton, ...
    vec_Jue_ton, ... % comentar Opcion B
    vec_Vie_ton, ...
    vec_ton_sem, ...
    vec_ton_anio, ...
    'VariableNames', { ...
        'Anio', ...
        'Factor', ...
        'Lunes_ton_dia', ...
        'Martes_ton_dia', ...
        'Miercoles_ton_dia', ...
        'Jueves_ton_dia', ...
        'Viernes_ton_dia', ...
    }

```

```

        'Ton_semana', ...
        'Ton_anio'}));

disp('=====');
disp('  Resumen de recepcion semanal por anio (ton/semana y ton/anio)');
disp('=====');
disp(tabla_recepcion);

%% =====
% SECCION 2: SIMULACION DIARIA - A2..A6
% =====

dias_totales    = n_anios * dias_por_anio;

% ----- A2: TOLVA -----
masa_recepcion_kg_dia    = zeros(dias_totales,1);
masa_salida_tolva_kg     = zeros(dias_totales,1);
lixiviado_tolva_L_dia    = zeros(dias_totales,1);
lixiviado_acum_tolva_L_dia = zeros(dias_totales,1);

% ----- A3: BANDA -----
masa_entrada_banda_kg_dia    = zeros(dias_totales,1);
masa_salida_banda_kg_dia     = zeros(dias_totales,1);
rechazo_banda_kg_dia         = zeros(dias_totales,1);
rechazo_banda_acum_kg_dia    = zeros(dias_totales,1);

% ----- A4: MOLINO -----
masa_entrada_molino_kg_dia    = zeros(dias_totales,1);
masa_salida_molino_kg_dia     = zeros(dias_totales,1);

% ----- A5: TORNILLO -----
masa_entrada_tornillo_kg_dia    = zeros(dias_totales,1);
masa_salida_tornillo_kg_dia     = zeros(dias_totales,1);

% ----- A6: CABAS -----
masa_salida_cabas_kg_dia        = zeros(dias_totales,1); % hacia patio
lixiviado_cabas_L_dia           = zeros(dias_totales,1); % L/dia cabas
masa_no_asignada_cabas_kg_dia   = zeros(dias_totales,1); % por falta de
capacidad
masa_total_en_cabas_kg_dia      = zeros(dias_totales,1);
fraccion_ocupacion_cabas_dia    = zeros(dias_totales,1);

```



```

% Para sumar líquidos de pirolisis al total de "líquidos"
% (suponemos densidad ~ 1 kg/L)
densidad_liquidos_piro_kg_L = 1.0;
liquidos_piro_L_dia = zeros(dias_totales,1);

% Masa en cada caba por día [kg]
masa_cabas_kg_dia = zeros(dias_totales, numero_cabas);
% Porcentaje de ocupacion por caba [%] (puede ser >100)
porcentaje_caba_dia = zeros(dias_totales, numero_cabas);

% Estado interno de las 6 cabas
caba_masa_inicial_kg = zeros(numero_cabas,1); % masa cargada al inicio [kg]
caba_dias_ocup = zeros(numero_cabas,1); % dias de retencion

% Para llenar simetricamente: índice de par actual (1: (1,2); 2: (3,4); 3: (5,6))
idx_par_siguiente = 1;

% ----- A7: PATIO DE SECADO -----
masa_salida_patio_kg_dia = zeros(dias_totales,1);
masa_no_asignada_patio_kg_dia = zeros(dias_totales,1);
lixiviado_patio_L_dia = zeros(dias_totales,1); % casi 0 pero lo
dejamos por compatibilidad

% Estado interno del patio (3 días FIFO)
patio_masa_dia = zeros(dias_retencion_patio,1);

% Para reporte mas detallado:
masa_patio_kg_dia = zeros(dias_totales,1);
porcentaje_ocup_patio_dia = zeros(dias_totales,1);

% ----- A8: PULVERIZADOR -----
masa_entrada_pulv_kg_dia = zeros(dias_totales,1);
masa_salida_pulv_kg_dia = zeros(dias_totales,1);

% ----- A9: ZARANDA / TROMEL -----
masa_entrada_zaranda_kg_dia = zeros(dias_totales,1);
masa_lista_horno_kg_dia = zeros(dias_totales,1); % salida final hacia el
horno

```

```

% ----- A10: HORNO DE PIROLISIS -----
masa_entrada_horno_kg_dia      = zeros(dias_totales,1);
masa_bioc_kg_dia                = zeros(dias_totales,1); % biocarbon (sólidos)
masa_liquidos_piro_kg_dia      = zeros(dias_totales,1); % líquidos de pirolisis
masa_gases_piro_kg_dia         = zeros(dias_totales,1); % gases de pirolisis

horas_horno_dia                 = zeros(dias_totales,1); % horas de operación del
horno

sacos_bioc_40kg_dia            = zeros(dias_totales,1); % número de sacos de 40
kg/dia

% ----- A11: CARGADOR FRONTAL -----
horas_cargador_dia = zeros(dias_totales,1);

%% =====
%   VECTORES DE HORAS DE OPERACION POR DIA (EQUIPOS ELECTRICOS)
% =====

horas_banda_dia      = zeros(dias_totales,1); % A3
horas_molino_dia     = zeros(dias_totales,1); % A4
horas_tornillo_dia    = zeros(dias_totales,1); % A5
horas_pulverizador_dia = zeros(dias_totales,1); % A8
horas_zaranda_dia     = zeros(dias_totales,1); % A9

% ----- bucle diario -----

for anio = 1:n_anios

    Lun_ton = vec_Lun_ton(anio);
    Mie_ton = vec_Mie_ton(anio);
    Vie_ton = vec_Vie_ton(anio);

    Mar_ton = vec_Mar_ton(anio);
    Jue_ton = vec_Jue_ton(anio);

    for dia = 1:dias_por_anio

        dia_global = (anio-1)*dias_por_anio + dia;

```

```

    semana = ceil(dia / 7);
    dia_semana = mod(dia-1, 7) + 1; % 1=lunes, 2=martes, ..., 7=domingo

ton_dia = 0;

if semana <= semanas_por_anio
    if dia_semana == 1      % Lunes
        ton_dia = Lun_ton;
    elseif dia_semana == 2 % Martes
        ton_dia = Mar_ton;
    elseif dia_semana == 3 % Miercoles
        ton_dia = Mie_ton;
    elseif dia_semana == 4 % Jueves
        ton_dia = Jue_ton;
    elseif dia_semana == 5 % Viernes
        ton_dia = Vie_ton;
    else
        ton_dia = 0;      % Sabado/Domingo
    end
end

% =====
% A2: TOLVA
% =====

masa_in_kg = ton_dia * 1000;
masa_recepcion_kg_dia(dia_global) = masa_in_kg;

lix_L = fraccion_escurrecimiento_tolva * masa_in_kg;
lixiviado_tolva_L_dia(dia_global) = lix_L;

masa_out_tolva_kg = masa_in_kg - lix_L;
masa_salida_tolva_kg(dia_global) = masa_out_tolva_kg;

if dia_global == 1
    lixiviado_acum_tolva_L_dia(dia_global) = lix_L;
else
    lixiviado_acum_tolva_L_dia(dia_global) = ...
        lixiviado_acum_tolva_L_dia(dia_global-1) + lix_L;
end

% =====
% A3: BANDA DE SELECCION
% =====

masa_in_banda_kg = masa_out_tolva_kg;

```

```

masa_entrada_banda_kg_dia(dia_global) = masa_in_banda_kg;

rechazo_kg = fraccion_rechazo_banda * masa_in_banda_kg;
rechazo_banda_kg_dia(dia_global) = rechazo_kg;

masa_out_banda_kg = masa_in_banda_kg - rechazo_kg;
masa_salida_banda_kg_dia(dia_global) = masa_out_banda_kg;

if dia_global == 1
    rechazo_banda_acum_kg_dia(dia_global) = rechazo_kg;
else
    rechazo_banda_acum_kg_dia(dia_global) = ...
        rechazo_banda_acum_kg_dia(dia_global-1) + rechazo_kg;
end

% =====
% A4: MOLINO TRITURADOR
% =====

masa_in_molino_kg = masa_out_banda_kg;
masa_entrada_molino_kg_dia(dia_global) = masa_in_molino_kg;

masa_out_molino_kg = masa_in_molino_kg;      % sin perdidas de masa
masa_salida_molino_kg_dia(dia_global) = masa_out_molino_kg;

% =====
% A5: TORNILLO SIN FIN
% =====

masa_in_tornillo_kg = masa_out_molino_kg;
masa_entrada_tornillo_kg_dia(dia_global) = masa_in_tornillo_kg;

masa_out_tornillo_kg = masa_in_tornillo_kg;  % sin perdidas de masa
masa_salida_tornillo_kg_dia(dia_global) = masa_out_tornillo_kg;

% =====
% HORAS DE OPERACION A3-A4-A5
% =====

% A3: banda de seleccion
if masa_in_banda_kg > 0
    horas_banda_dia(dia_global) = masa_in_banda_kg / vel_banda_kg_h;
else
    horas_banda_dia(dia_global) = 0;
end

```

```

% A4: molino triturador
if masa_in_molino_kg > 0
    horas_molino_dia(dia_global) = masa_in_molino_kg / vel_molino_kg_h;
else
    horas_molino_dia(dia_global) = 0;
end

% A5: tornillo sin fin
if masa_in_tornillo_kg > 0
    horas_tornillo_dia(dia_global) = masa_in_tornillo_kg /
vel_tornillo_kg_h;
else
    horas_tornillo_dia(dia_global) = 0;
end

% =====
% A6: CABAS DE DRENAJE
% =====

% 1) Actualizar dias de retencion y liberar cabas que cumplen 3 dias
lix_cabas_hoy_L = 0;
masa_salida_cabas_hoy_kg = 0;

for i = 1:numero_cabas
    if caba_masa_inicial_kg(i) > 0
        caba_dias_ocup(i) = caba_dias_ocup(i) + 1;
        if caba_dias_ocup(i) >= dias_retencion_cabas
            M_in_caba = caba_masa_inicial_kg(i);
            if M_in_caba > 0
                M_out_caba = (1 - fraccion_perdida_caba) * M_in_caba;
                L_caba      = lixiviado_por_kg_caba_Lkg * M_in_caba;

                masa_salida_cabas_hoy_kg = masa_salida_cabas_hoy_kg +
M_out_caba;
                lix_cabas_hoy_L          = lix_cabas_hoy_L          +
L_caba;

                % Liberar caba
                caba_masa_inicial_kg(i) = 0;
                caba_dias_ocup(i)        = 0;
            end
        end
    end
end
end
end

```

```

masa_salida_cabas_kg_dia(dia_global) = masa_salida_cabas_hoy_kg;
lixiviado_cabas_L_dia(dia_global)    = lix_cabas_hoy_L;

% 2) Asignar nueva masa desde tornillo a las cabas (llenado simetrico)
masa_disponible_cabas_kg = masa_salida_tornillo_kg_dia(dia_global);
masa_no_asignada = masa_disponible_cabas_kg;

if masa_disponible_cabas_kg > 0
    intentos = 0;
    while masa_no_asignada > 0 && intentos < 3    % max 3 pares
        intentos = intentos + 1;

        % Seleccion del par actual
        if idx_par_siguiente == 1
            cabas_par = [1 2];
        elseif idx_par_siguiente == 2
            cabas_par = [3 4];
        else
            cabas_par = [5 6];
        end

        % Verificar si ambas cabas del par estan libres
        if all(caba_masa_inicial_kg(cabas_par) == 0)
            n_cabas_par = numel(cabas_par);
            masa_teorica_cada = masa_no_asignada / n_cabas_par;

            % NO limitamos a masa_max_caba_kg: permitimos >100 % de
ocupacion

            masa_cada = masa_teorica_cada;

            if masa_cada > 0
                for j = cabas_par
                    caba_masa_inicial_kg(j) = masa_cada;
                    caba_dias_ocup(j)      = 0;
                end
                masa_asignada      = masa_cada * n_cabas_par;
                masa_no_asignada = masa_no_asignada - masa_asignada;
            end

            % Avanzar al siguiente par
            idx_par_siguiente = idx_par_siguiente + 1;
            if idx_par_siguiente > 3
                idx_par_siguiente = 1;
            end
        else
            % Par ocupado, probar siguiente
            idx_par_siguiente = idx_par_siguiente + 1;
            if idx_par_siguiente > 3

```

```

        idx_par_siguiente = 1;
    end
end
end
end

masa_no_asignada_cabas_kg_dia(dia_global) = masa_no_asignada;

% 3) Ocupacion total de cabas
masa_total_cabas_kg = sum(caba_masa_inicial_kg);
masa_total_en_cabas_kg_dia(dia_global) = masa_total_cabas_kg;
fraccion_ocupacion_cabas_dia(dia_global) = ...
    masa_total_cabas_kg / (masa_max_caba_kg * numero_cabas);

% Guardar masa por caba y porcentaje diario
masa_cabas_kg_dia(dia_global, :) = caba_masa_inicial_kg';
porcentaje_caba_dia(dia_global, :) = (caba_masa_inicial_kg' /
masa_max_caba_kg) * 100;

% =====
% A7: PATIO DE SECADO (3 días de retención)
% =====

% 1) Liberar el material que ya cumplió los 3 días
masa_salida_hoy_patio = patio_masa_dia(end);
patio_masa_dia(2:end) = patio_masa_dia(1:end-1);
patio_masa_dia(1) = 0;

if masa_salida_hoy_patio > 0
    % aplicar pérdida del patio
    masa_post_patio = (1 - fraccion_perdida_patio) *
masa_salida_hoy_patio;

    masa_salida_patio_kg_dia(dia_global) = masa_post_patio;

    % Patio no genera lixiviado medible (datos experimentales)
    lixiviado_patio_L_dia(dia_global) = 0;
end

% 2) Ingresar nueva biomasa desde cabas
masa_entrada_patio = masa_salida_cabas_kg_dia(dia_global);

```

```

    % No limitamos la ocupación → si se pasa del 100 %, significa mayor
    espesor real
    patio_masa_dia(1) = patio_masa_dia(1) + masa_entrada_patio;

    % 3) Guardar estado del patio
    masa_total_patio = sum(patio_masa_dia);
    masa_patio_kg_dia(dia_global) = masa_total_patio;

    porcentaje_ocup_patio_dia(dia_global) = ...
        (masa_total_patio / masa_max_patio_kg) * 100;

    % 4) Si hubiera falta de capacidad real (solo ocurriría si lo
    bloqueamos),
    % dejamos este contador preparado para análisis:
    masa_no_asignada_patio_kg_dia(dia_global) = 0;

    % =====
    % A8: PULVERIZADOR DE BIOMASA
    % =====

    % Entrada al pulverizador: lo que sale del patio hoy
    masa_in_pulv = masa_salida_patio_kg_dia(dia_global);
    masa_entrada_pulv_kg_dia(dia_global) = masa_in_pulv;

    % No hay perdidas de masa en el pulverizador (solo cambio de
    granulometria)
    masa_out_pulv = masa_in_pulv;
    masa_salida_pulv_kg_dia(dia_global) = masa_out_pulv;

    % =====
    % A9: ZARANDA / TROMEL
    % =====

    % Entrada a la zaranda: salida del pulverizador
    masa_in_zaranda = masa_out_pulv;
    masa_entrada_zaranda_kg_dia(dia_global) = masa_in_zaranda;

    % Fraccion gruesa recircula hasta ser apta: en el balance neto
    % del dia no hay perdida de masa
    masa_out_zaranda = masa_in_zaranda;

```



```

% Esta masa es la que queda "lista para horno"
masa_lista_horno_kg_dia(dia_global) = masa_out_zaranda;

% =====
% HORAS DE OPERACION A8-A9 (por velocidad de proceso)
% =====

% A8: pulverizador
if masa_in_pulv > 0
    horas_pulverizador_dia(dia_global) = masa_in_pulv /
vel_pulverizador_kg_h;
else
    horas_pulverizador_dia(dia_global) = 0;
end

% A9: zaranda / tromel
if masa_in_zaranda > 0
    horas_zaranda_dia(dia_global) = masa_in_zaranda / vel_zaranda_kg_h;
else
    horas_zaranda_dia(dia_global) = 0;
end

% =====
% A10: HORNO DE PIROLISIS
% =====

% Masa que entra al horno hoy (kg)
masa_in_horno = masa_lista_horno_kg_dia(dia_global);
masa_entrada_horno_kg_dia(dia_global) = masa_in_horno;

if masa_in_horno > 0

    % Horas de operacion del horno, usando capacidad nominal [kg/h]
    horas_horno_dia(dia_global) = masa_in_horno / capacidad_horno_kg_h;

    % Rendimientos interpolados para la temperatura de operación
T_horno_C
    fr_solid = interp1(T_exp, R_solid_fr, T_horno_C, 'linear',
'extrap');

```

```

fr_liquid = interp1(T_exp, R_liquid_fr, T_horno_C, 'linear',
'extrap');
fr_gas     = interp1(T_exp, R_gas_fr,    T_horno_C, 'linear',
'extrap');

% Normalizamos por si la suma no es exactamente 1
suma_fr = fr_solid + fr_liquid + fr_gas;
if suma_fr > 0
    fr_solid = fr_solid / suma_fr;
    fr_liquid = fr_liquid / suma_fr;
    fr_gas    = fr_gas    / suma_fr;
end

% Masas de productos de pirólisis [kg/dia]
m_solid = fr_solid * masa_in_horno;
m_liquid = fr_liquid * masa_in_horno;
m_gas    = fr_gas    * masa_in_horno;

masa_bioc_kg_dia(dia_global)      = m_solid;
masa_liquidos_piro_kg_dia(dia_global) = m_liquid;
masa_gases_piro_kg_dia(dia_global)  = m_gas;

% Sacos de 40 kg de biocarbon
sacos_bioc_40kg_dia(dia_global) = m_solid / 40;

% Liquidos de pirolisis en litros (para sumarlos a "liquidos
totales")
liquidos_piro_L_dia(dia_global) = m_liquid /
densidad_liquidos_piro_kg_L;

else
    horas_horno_dia(dia_global)      = 0;
    masa_bioc_kg_dia(dia_global)      = 0;
    masa_liquidos_piro_kg_dia(dia_global) = 0;
    masa_gases_piro_kg_dia(dia_global)  = 0;
    sacos_bioc_40kg_dia(dia_global)    = 0;
    liquidos_piro_L_dia(dia_global)    = 0;
end

% =====
% A11: CARGADOR FRONTAL - HORAS DE OPERACION
% =====

```

```

% 1) Determinar si hoy hay operacion de maxima intensidad
tiene_mov_cabas = (masa_salida_cabas_kg_dia(dia_global) > 0);
tiene_mov_patio = (masa_salida_patio_kg_dia(dia_global) > 0);
tiene_mov_horno = (masa_entrada_horno_kg_dia(dia_global) > 0);

dia_maxima_oper = (tiene_mov_cabas || tiene_mov_patio ||
tiene_mov_horno);

% 2) Determinar si, sin ser dia de maxima, hay biomasa en cabas o patio
masa_total_cabas_hoy = sum(masa_cabas_kg_dia(dia_global, :));
masa_patio_hoy      = masa_patio_kg_dia(dia_global);

dia_minima_oper = (~dia_maxima_oper) && ...
(masa_total_cabas_hoy > 0 || masa_patio_hoy > 0);

% 3) Calcular masa movida intensamente (solo para escalar dias de maxima)
masa_mov_intensa = ...
    masa_salida_cabas_kg_dia(dia_global) ...
    + masa_salida_patio_kg_dia(dia_global) ...
    + masa_entrada_horno_kg_dia(dia_global);

if dia_maxima_oper
    % Factor de intensidad relativo a la referencia 4 t/dia
    if masa_ref_intensa_kg_dia > 0
        factor_intensidad = masa_mov_intensa / masa_ref_intensa_kg_dia;
    else
        factor_intensidad = 1.0;
    end

    % Horas base escaladas por la intensidad de masa
    horas_max_hoy = H_carg_max_base_h_dia * factor_intensidad;

    % Limitar a un tope razonable
    horas_cargador_dia(dia_global) = min(horas_max_hoy,
H_carg_max_tope_h_dia);

elseif dia_minima_oper
    % Dias con biomasa en cabas/patio pero sin mov grande
    horas_cargador_dia(dia_global) = H_carg_min_h_dia;

```

```

        else
            % Sin biomasa relevante ni movimientos -> sin uso del cargador
            horas_cargador_dia(dia_global) = 0;
        end
    end
end

%% -----
% 5. REVISION RAPIDA DE CONSISTENCIA
% -----

total_ton_tabla = sum(vec_ton_anio);
total_ton_dias = sum(masa_recepcion_kg_dia) / 1000;

total_lix_tolva_L = lixiviado_acum_tolva_L_dia(end);
total_rechazo_banda_ton = rechazo_banda_acum_kg_dia(end) / 1000;

total_masa_salida_tornillo_ton = sum(masa_salida_tornillo_kg_dia) / 1000;
total_masa_salida_cabas_ton = sum(masa_salida_cabas_kg_dia) / 1000;
total_lix_cabas_L = sum(lixiviado_cabas_L_dia);
total_masa_no_asignada_cabas_ton = sum(masa_no_asignada_cabas_kg_dia) / 1000;
total_masa_salida_patio_ton = sum(masa_salida_patio_kg_dia) / 1000;
total_masa_lista_horno_ton = sum(masa_lista_horno_kg_dia) / 1000;
total_masa_bioc_ton = sum(masa_bioc_kg_dia) / 1000;
total_sacos_bioc = sum(sacos_bioc_40kg_dia);
total_liquidos_piro_L = sum(liquidos_piro_L_dia);
total_masa_gases_piro_ton = sum(masa_gases_piro_kg_dia) / 1000;
lixiviado_total_L = total_lix_tolva_L + total_lix_cabas_L +
total_liquidos_piro_L;

fprintf('\n===== REVISION DE CONSISTENCIA =====\n');
fprintf('Total segun tabla anual: %.2f ton\n', total_ton_tabla);
fprintf('Total recepcion sumando dias: %.2f ton\n', total_ton_dias);
fprintf('Diferencia (tabla - dias): %.4f ton\n', total_ton_tabla -
total_ton_dias);
fprintf('Lixiviado total en tolva (A2): %.2f L en 15 años\n',
total_lix_tolva_L);
fprintf('Rechazo total en banda (A3): %.2f ton en 15 años\n',
total_rechazo_banda_ton);
fprintf('Masa total a la salida del tornillo (A5): %.2f ton\n',
total_masa_salida_tornillo_ton);

```

```

fprintf('Masa total que sale de cabas (A6):          %.2f ton\n',
total_masa_salida_cabas_ton);
fprintf('Lixiviado total en cabas (A6):            %.2f L\n', total_lix_cabas_L);
fprintf('Masa NO asignada a cabas (cuello botella): %.2f ton\n',
total_masa_no_asignada_cabas_ton);
fprintf('Masa total que sale del patio (A7):         %.2f ton en 15 años\n',
total_masa_salida_patio_ton);
fprintf('Masa total lista para horno (A9):           %.2f ton en 15 años\n',
total_masa_lista_horno_ton);
fprintf('Masa total de biocarbon (A10):               %.2f ton en 15 años\n',
total_masa_bioc_ton);
fprintf('Numero total de sacos de 40 kg:              %.0f sacos\n',
total_sacos_bioc);
fprintf('Liquidos de pirolisis (A10):                 %.2f L en 15 años\n',
total_liquidos_piro_L);
fprintf('Liquidos totales (tolva + cabas + pirólisis): %.2f L\n',
lixiviado_total_L);
fprintf('Masa total de gases de pirolisis (A10): %.2f ton en 15 años\n',
total_masa_gases_piro_ton);

fprintf('=====\\n');

%% 6. DEFINIR VENTANA DE LOS ULTIMOS 30 DIAS DEL ANIO 15
dia_final = dias_totales;      % ultimo dia simulado (fin del año 15)
dia_ini   = dia_final - 29;    % ultimos 30 días

% %6. DEFINIR VENTANA de días que quiero ver en tablas
% dia_final = 30;
% dia_ini   = 1;

dias_ventana = dia_ini:dia_final;

%% 7. TABLA - PATIO: ULTIMOS 30 DIAS DEL ANIO 15

tabla_patio_ultimos30 = table( ...
    (dia_ini:dia_final)', ...
    masa_patio_kg_dia(dia_ini:dia_final), ...
    porcentaje_ocup_patio_dia(dia_ini:dia_final), ...
    'VariableNames', {'Dia_global', 'MasaPatio_kg', 'Ocupacion_porc'});

disp('-----');
disp('Patio de secado - ultimos 30 días del año 15');

```

```

disp('-----');
disp(tabla_patio_ultimos30);

%% 8. TABLA - CABAS (MASA EN kg) ULTIMOS 30 DIAS DEL ANIO 15

masa_cabas_kg_ultimos30 = masa_cabas_kg_dia(dias_ventana, :); % [30 x 6]

tabla_cabas_kg_ultimos30 = table( ...
    dias_ventana', ...
    masa_cabas_kg_ultimos30(:,1), ...
    masa_cabas_kg_ultimos30(:,2), ...
    masa_cabas_kg_ultimos30(:,3), ...
    masa_cabas_kg_ultimos30(:,4), ...
    masa_cabas_kg_ultimos30(:,5), ...
    masa_cabas_kg_ultimos30(:,6), ...
    'VariableNames', {'Dia_global', ...
        'Caba1_kg', 'Caba2_kg', 'Caba3_kg', ...
        'Caba4_kg', 'Caba5_kg', 'Caba6_kg'});

disp('-----');
disp('Cabas - masa (kg) ultimos 30 dias del anio 15');
disp('-----');
disp(tabla_cabas_kg_ultimos30);

%% 9. TABLA - CABAS (OCUPACION %) ULTIMOS 30 DIAS DEL ANIO 15

ocup_cabas_porc_ultimos30 = porcentaje_caba_dia(dias_ventana, :); % [30 x 6]

tabla_cabas_ocup_ultimos30 = table( ...
    dias_ventana', ...
    ocup_cabas_porc_ultimos30(:,1), ...
    ocup_cabas_porc_ultimos30(:,2), ...
    ocup_cabas_porc_ultimos30(:,3), ...
    ocup_cabas_porc_ultimos30(:,4), ...
    ocup_cabas_porc_ultimos30(:,5), ...
    ocup_cabas_porc_ultimos30(:,6), ...
    'VariableNames', {'Dia_global', ...
        'Caba1_pct', 'Caba2_pct', 'Caba3_pct', ...
        'Caba4_pct', 'Caba5_pct', 'Caba6_pct'});

disp('-----');

```

```

disp('Cabas - ocupacion (%) ultimos 30 dias del anio 15');
disp('-----');
disp(tabla_cabas_ocup_ultimos30);

%% 10. TABLA RESUMEN ANUAL - MASAS, RENDIMIENTO Y LIXIVIADOS (SEPARADOS)

% Preallocar (TODAS como columna n_anios x 1)
masa_recibida_ton_anual      = zeros(n_anios,1);
masa_entrada_horno_ton_anual = zeros(n_anios,1);
biocarbon_ton_anual          = zeros(n_anios,1);
rendimiento_global_pct_anual = zeros(n_anios,1);

lixiviado_real_L_anual       = zeros(n_anios,1); % tolva + cabas + patio
liquidos_piro_L_anual        = zeros(n_anios,1); % líquidos de pirólisis
(bio-oil/condensados)

for a = 1:n_anios
    idx_ini = (a-1)*dias_por_anio + 1;
    idx_fin = a * dias_por_anio;
    idx      = idx_ini:idx_fin;

    % Masa recibida en tolva (entrada al sistema) [ton/año]
    masa_recibida_ton_anual(a) = sum(masa_recepcion_kg_dia(idx)) / 1000;

    % Masa total que entra al horno [ton/año]
    masa_entrada_horno_ton_anual(a) = sum(masa_entrada_horno_kg_dia(idx)) / 1000;

    % Biocarbon producido [ton/año]
    biocarbon_ton_anual(a) = sum(masa_bioc_kg_dia(idx)) / 1000;

    % Rendimiento global a biocarbon (respecto a lo recibido en tolva) [%]
    if masa_recibida_ton_anual(a) > 0
        rendimiento_global_pct_anual(a) = 100 * biocarbon_ton_anual(a) /
masa_recibida_ton_anual(a);
    else
        rendimiento_global_pct_anual(a) = 0;
    end

    % Lixiviado real (drenajes) [L/año]
    lixiviado_real_L_anual(a) = ...

```

```

        sum(lixiviado_tolva_L_dia(idx)) ...
    + sum(lixiviado_cabas_L_dia(idx)) ...
    + sum(lixiviado_patio_L_dia(idx));

    % Líquidos de pirólisis [L/año]
    liquidos_piro_L_anual(a) = sum(liquidos_piro_L_dia(idx));
end

% Forzar TODO a columna (por seguridad)
Anio_col = (1:n_anios)';

masa_recibida_ton_anual      = masa_recibida_ton_anual(:);
masa_entrada_horno_ton_anual = masa_entrada_horno_ton_anual(:);
biocarbon_ton_anual          = biocarbon_ton_anual(:);
rendimiento_global_pct_anual = rendimiento_global_pct_anual(:);
lixiviado_real_L_anual        = lixiviado_real_L_anual(:);
liquidos_piro_L_anual         = liquidos_piro_L_anual(:);

% Construir tabla
tabla_resumen_masas_anual = table( ...
    Anio_col, ...
    masa_recibida_ton_anual, ...
    masa_entrada_horno_ton_anual, ...
    biocarbon_ton_anual, ...
    rendimiento_global_pct_anual, ...
    lixiviado_real_L_anual, ...
    liquidos_piro_L_anual, ...
    'VariableNames', { ...
        'Anio', ...
        'Masa_recibida_ton', ...
        'Masa_entrada_horno_ton', ...
        'Biocarbon_ton', ...
        'Rend_global_biocarb_pct', ...
        'Lixiviado_real_L', ...
        'Liquidos_piro_L'});

disp('-----');
disp('Tabla resumen anual - masas, rendimiento y lixiviados (separados)');
disp('-----');
disp(tabla_resumen_masas_anual);

% Chequeo rápido de ratios (con protección división por cero)

```



```

den_rec_kg    = tabla_resumen_masas_anual.Masa_recibida_ton * 1000;
den_horno_kg = tabla_resumen_masas_anual.Masa_entrada_horno_ton * 1000;

ratio_lix_real = tabla_resumen_masas_anual.Lixiviado_real_L ./ max(den_rec_kg,
1e-12);
ratio_piro     = tabla_resumen_masas_anual.Liquidos_piro_L    ./ max(den_horno_kg,
1e-12);

fprintf('Lixiviado real [L/kg recibido] año 1: %.6f\n', ratio_lix_real(1));
fprintf('Liquidos piro  [L/kg al horno]  año 1: %.6f\n', ratio_piro(1));

%% 11. TABLA RESUMEN ANUAL - OPERACION DEL HORNO

sacos_bioc_anual      = zeros(n_anios,1);
liquidos_piro_L_anual = zeros(n_anios,1);
gases_piro_ton_anual  = zeros(n_anios,1);
horas_horno_anual     = zeros(n_anios,1);
horas_horno_sem_anual = zeros(n_anios,1);

semanas_operacion = 52; % semanas operativas por año

for a = 1:n_anios
    idx_ini = (a-1)*dias_por_anio + 1;
    idx_fin = a    *dias_por_anio;
    idx     = idx_ini:idx_fin;

    sacos_bioc_anual(a)      = sum(sacos_bioc_40kg_dia(idx));
    liquidos_piro_L_anual(a) = sum(liquidos_piro_L_dia(idx));
    gases_piro_ton_anual(a)  = sum(masa_gases_piro_kg_dia(idx)) / 1000;
    horas_horno_anual(a)     = sum(horas_horno_dia(idx));

    % Horas promedio de horno por semana [h/sem]
    horas_horno_sem_anual(a) = horas_horno_anual(a) / semanas_operacion;
end

tabla_horno_anual = table( ...
    (1:n_anios)', ...
    horas_horno_sem_anual, ...
    horas_horno_anual, ...
    sacos_bioc_anual, ...

```

```

liquidos_piro_L_anual, ...
gases_piro_ton_anual, ...
'VariableNames', { ...
    'Anio', ...
    'Horas_horno_prom_sem', ... % h/semana
    'Horas_horno_anio', ...     % h/año
    'Sacos_40kg', ...
    'Liquidos_piro_L', ...
    'Gases_piro_ton'}});

disp('-----');
disp('Tabla resumen anual - operación del horno');
disp('-----');
disp(tabla_horno_anual);

%% 12. TABLA - HORNO: ULTIMOS 30 DIAS DEL ANIO 15
tabla_horno_ultimos30 = table( ...
    dias_ventana', ...
    masa_entrada_horno_kg_dia(dias_ventana), ...
    horas_horno_dia(dias_ventana), ...
    masa_bioc_kg_dia(dias_ventana), ...
    sacos_bioc_40kg_dia(dias_ventana), ...
    liquidos_piro_L_dia(dias_ventana), ...
    masa_gases_piro_kg_dia(dias_ventana), ...
    'VariableNames', { ...
        'Dia_global', ...
        'Masa_in_horno_kg', ...
        'Horas_horno', ...
        'Biocarbon_kg', ...
        'Sacos_40kg', ...
        'Liquidos_piro_L', ...
        'Gases_piro_kg'}});

disp('-----');
disp('Horno - ultimos 30 dias del anio 15 (regimen)');
disp('-----');
disp(tabla_horno_ultimos30);

%% 13. TABLA RESUMEN ANUAL - OPERACION DEL CARGADOR (A11)

horas_cargador_anual = zeros(n_anios,1);
horas_cargador_sem_anual = zeros(n_anios,1);

```

```

semanas_operacion = 52; % ya usado antes

for a = 1:n_anios
    idx_ini = (a-1)*dias_por_anio + 1;
    idx_fin = a * dias_por_anio;
    idx = idx_ini:idx_fin;

    horas_cargador_anual(a) = sum(horas_cargador_dia(idx));
    horas_cargador_sem_anual(a) = horas_cargador_anual(a) / semanas_operacion;
end

tabla_cargador_anual = table( ...
    (1:n_anios)', ...
    horas_cargador_sem_anual, ...
    horas_cargador_anual, ...
    'VariableNames', { ...
        'Anio', ...
        'Horas_cargador_prom_sem', ... % h/semana
        'Horas_cargador_anio'}); % h/año

disp('-----');
disp('Tabla resumen anual - operacion del cargador');
disp('-----');
disp(tabla_cargador_anual);

%% =====
% LLAMAR MODELO ENERGETICO
% =====

% Construimos el struct "datos" con las horas diarias de cada equipo.
% Estas variables ya existen en el script principal.
datos = struct();
datos.horas_A3_dia = horas_banda_dia; % A3
datos.horas_A4_dia = horas_molino_dia; % A4
datos.horas_A5_dia = horas_tornillo_dia; % A5
datos.horas_A8_dia = horas_pulverizador_dia; % A8
datos.horas_A9_dia = horas_zaranda_dia; % A9
datos.horas_horno_dia = horas_horno_dia; % A10
datos.horas_cargador_dia = horas_cargador_dia; % A11

% --- NUEVO: para GN + KPI por producto ---
datos.masa_entrada_horno_kg_dia = masa_entrada_horno_kg_dia; % biomasa al horno

```

```

datos.masa_bioc_kg_dia      = masa_bioc_kg_dia;          % biocarbón
producido
datos.sacos_40kg_dia       = sacos_bioc_40kg_dia;        % sacos de 40 kg

% Masa diaria al horno (necesaria para energia termica del quemador)
datos.masa_entrada_horno_kg_dia = masa_entrada_horno_kg_dia;

% Humedad de alimentacion al horno (usamos la humedad final del patio como
aproximacion)
datos.hum_in_horno_frac = humedad_final_patio;    % ej. 0.1034 (base húmeda)

% (Opcional) modelar auxiliares y administrativos:
% datos.horas_A12_aux_dia = ... (vector dias_totales x 1)
% datos.horas_admin_dia   = ... (vector dias_totales x 1)

% Ejecutar el modelo energético
res_ener = modelo_energetico_planta(param, datos);

disp('=====');
disp('ELECTRICIDAD ANUAL POR EQUIPO (kWh/año)');
disp('=====');
disp(res_ener.tabla_elec_por_equipo_anual);

disp('=====');
disp('RESULTADOS TERMODINAMICOS DEL HORNO (QUEMADOR GN) - RESUMEN ANUAL');
disp('=====');
disp(res_ener.tabla_quemador_anual);

disp('=====');
disp('RESUMEN ENERGIA ANUAL (kWh) - ELECTRICO + DIESEL + GN');
disp('=====');
disp(res_ener.tabla_energia_anual);

disp('=====');
disp('KPI: kWh POR TON DE BIOCARBON (PRODUCTO)');
disp('=====');
disp(res_ener.tabla_kpi_kWh_ton_bioc);

disp('=====');
disp('KPI: kWh POR SACO DE 40 kg (PRODUCTO)');

```

```

disp('=====');
disp(res_ener.tabla_kpi_kWh_saco_40kg);

%% -----
%  PARAMETROS MODELO_EXERGETICO_PLANTA
% -----

param_ex = struct();

% LHV/PCI en MJ (consistencia con exergía)
param_ex.PCI_GN_MJ_m3 = 3.6 * param.PCI_GN_kWh_m3;    % MJ/Nm3

% Factores beta (literatura)
param_ex.beta_GN = 1.04;    % gas natural
param_ex.beta_bio = 1.12;    % biomasa (correlación H/C y O/C segun Song)
param_ex.beta_BC = 1.08;    % biocarbón (correlación H/C y O/C segun Song)
param_ex.LHV_liq_MJ_kg = 7;    % rango Basu: 5-10 MJ/kg
param_ex.beta_liq = 1.10;    % correlación Elhenawy, H/C y O/C segun Basu
param_ex.LHV_gas_MJ_Nm3 = 6;    % rango Basu: 4-7 MJ/Nm3
param_ex.beta_gas = 1.02;    % syngas pobre, baja T
param_ex.rho_gas_piro_kg_Nm3 = 1.2;    % [kg/Nm3] syngas pobre

% PCI/LHV biomasa y biocarbón [MJ/kg]
param_ex.LHV_bio_MJ_kg = 14.3;
param_ex.LHV_BC_MJ_kg = 16.7;

% cp productos pirólisis [J/kg-K] (promedios)
param_ex.cp_BC_J_kgK = 1000;    %
param_ex.cp_liq_J_kgK = 2200;    %
param_ex.cp_gas_J_kgK = 2000;    %

% Temperaturas (si no pones, el módulo las toma de param)
% param_ex.T0_K = param.T_amb_C + 273.15;
% param_ex.T_fg_K = param.T_fg_C + 273.15;
% param_ex.T_proc_K= param.T_proc_C+ 273.15;

%% -----
%  LLAMAR MODELO_EXERGETICO (HORNO )
% -----

datos_ex = struct();
datos_ex.m_in_horno_kg_dia = masa_entrada_horno_kg_dia;
datos_ex.m_BC_kg_dia = masa_bioc_kg_dia;
datos_ex.m_liq_kg_dia = masa_liquidos_piro_kg_dia;

```

```

datos_ex.m_gas_kg_dia      = masa_gases_piro_kg_dia;

% Electricidad del horno: unimos soplador + auxiliares

E_soplador_kWh_dia  = (param.P_soplador_horno_kW / param.eta_soplador) *
horas_horno_dia;
E_aux_horno_kWh_dia = (param.P_aux_horno_kW      / param.eta_aux_horno) *
horas_horno_dia;

datos_ex.E_el_horno_kWh_dia = E_soplador_kWh_dia + E_aux_horno_kWh_dia;

% Humedad entrada al horno
datos_ex.hum_in_horno_frac = datos.hum_in_horno_frac;

% Ejecutar exergía
res_ex = modelo_exergetico_planta(param_ex, param, res_ener, datos_ex);

disp('=====');
disp('EXERGIA HORNO (A10) - EXERGIA DESTRUIDA Y EFICIENCIA (ANUAL, MJ)');
disp('=====');
disp(res_ex.tabla_ex_horno_anual);

disp('=====');
disp('EXERGIA HORNO (A10) - EXERGIA DESTRUIDA Y EFICIENCIA (MJ/kg_in)');
disp('=====');
disp(res_ex.tabla_ex_horno_MJkg_in);

%% =====
%  MODELO ECONOMICO (USD) - INGRESOS + OPEX + CAPEX + KPI FINANCIEROS
% =====
%  Escenario base conservador - horizonte (15 años)  TRM ~4000
% ----- PARAMETROS ECONOMICOS -----
param_econ = struct();

% Horizonte
param_econ.n_anios = n_anios;

% CAPEX (año 0) Inversión total estimada
param_econ.capex_USD = 711955;

```

```

% Tasa de descuento costo de oportunidad del capital y el riesgo del proyecto
param_econ.tasa_descuento = 0.09; % 9% anual

% ----- Ingresos (año 1) -----
param_econ.precio_bulto_USD = 25; % USD / bulto 40 kg
param_econ.ahorro_disp_USD_ton = 30; % USD / ton recibida
param_econ.precio_lix_USD_L = 0; % USD / L lixiviado (no considerado en
escenario base)

% Crecimientos anuales real moderado: 3%
param_econ.g_bulto = 0.03;
param_econ.g_ahorro = 0.03;
param_econ.g_lix = 0.0;

% ----- Precios energía (año 1) + incrementos -----
param_econ.p_elec_USD_kWh_1 = 0.2; % USD/kWh (tarifa promedio)
param_econ.p_diesel_USD_L_1 = 0.7; % USD/L basado en ACPM ~10.925 COP/gal
(2026)
param_econ.p_gn_USD_m3_1 = 0.5; % USD/m3 (valor base ajustable segun
negociacion)

param_econ.g_elec = 0.05;
param_econ.g_diesel = 0.05;
param_econ.g_gn = 0.05;

% ----- OPEX fijo anual -----
% Mano de obra
param_econ.salario_USD_mes = 650; % SMLV a 2026 (salario + prestaciones +
cargas)
param_econ.n_trabajadores = 5; % 3 operarios + 1 administrativo + 1 jefe de
planta
                                %(B)+ 2 operarios +1 administrativos + 2
comerciales
param_econ.g_salario = 0.06; % Refleja inflación y ajustes históricos

% Mantenimiento y repuestos: 4% de valor equipos
param_econ.mantto_USD_anual = 5310;

% Insumos y materiales:1% del CAPEX nueva / 6% del CAPEX Existente
param_econ.frac_insumos = 0.01;

```

```
% Otros gastos: 1% del OPEX total
param_econ.frac_otros = 0.01;

% Para convertir diésel kWh->L (desde tu param energético)
param_econ.PCI_diesel_kWh_L = param.PCI_diesel_kWh_L;

% ----- EJECUTAR MODELO ECONOMICO -----
res_econ = modelo_economico_planta(param_econ, res_ener,
tabla_resumen_masas_anual);

disp('=====');
disp('TABLA ECONOMICA FINAL (AÑO 0..n) - COHERENCIA');
disp('=====');
disp(res_econ.tabla_economica_0n);

disp('=====');
disp('INDICADORES FINANCIEROS');
disp('=====');
disp(res_econ.resumen_financiero);

disp('=====');
disp('PRECIO MINIMO DEL BULTO (USD/bulto 40 kg)');
disp('=====');
disp(res_econ.resumen_precio_minimo);
```



```

function res_ener = modelo_energetico_planta(param, datos)
% MODELO_ENERGETICO_PLANTA
% -----
% Modelo energetico 0D de la planta (electricidad + diesel + GN del quemador).
% incluyendo calor de gases de escape y pérdidas (frac_perdidas).
% Incluye estequiometría CH4 + aire con exceso (lambda_air).
% -----

%% 0) VALIDACION PARAMETROS REQUERIDOS
campos_param = { ...
    'P_A3_kW', 'eta_A3', ...
    'P_A4_kW', 'eta_A4', ...
    'P_A5_kW', 'eta_A5', ...
    'P_A8_kW', 'eta_A8', ...
    'P_A9_kW', 'eta_A9', ...
    'P_soplador_horno_kW', 'eta_soplador', ...
    'P_aux_horno_kW', 'eta_aux_horno', ...
    'P_A12_aux_kW', 'eta_A12_aux', ...
    'P_admin_kW', 'eta_admin', ...
    'PCI_GN_kWh_m3', 'PCI_diesel_kWh_L', ...
    'consumo_diesel_L_h', ...
    'n_anios', 'dias_por_anio', ...
    ... % horno/balance
    'T_in_C', 'T_proc_C', 'T_amb_C', 'T_fg_C', ...
    'frac_perdidas', 'lambda_air', ...

    'cp_bio_J_kgK', 'cp_w_liq_J_kgK', 'cp_w_vap_J_kgK', 'h_vap_J_kg', 'Q_torref_J_kg_dry'
, ...
    ... % cp gases (molar)
    'cp_CO2_kJ_kmolK', 'cp_H2O_kJ_kmolK', 'cp_O2_kJ_kmolK', 'cp_N2_kJ_kmolK',
    ...
    ... % masas molares
    'M_CO2', 'M_H2O', 'M_O2', 'M_N2' ...
};

for i = 1:numel(campos_param)
    if ~isfield(param, campos_param{i})
        error('modelo_energetico_planta: falta param. %s', campos_param{i});
    end
end

%% 1) VALIDACION DATOS REQUERIDOS
campos_datos = { ...

    'horas_A3_dia', 'horas_A4_dia', 'horas_A5_dia', 'horas_A8_dia', 'horas_A9_dia', ...

```

```

        'horas_horno_dia', 'horas_cargador_dia', ...
        'masa_entrada_horno_kg_dia', 'masa_bioc_kg_dia', 'sacos_40kg_dia', ...
        'hum_in_horno_frac' ...
    };

    for i = 1:numel(campos_datos)
        if ~isfield(datos, campos_datos{i})
            error('modelo_energetico_planta: falta datos.%.s', campos_datos{i});
        end
    end

    %% 2) PARAMETROS TEMPORALES
    n_anios      = param.n_anios;
    dias_por_anio = param.dias_por_anio;
    dias_total_esp = n_anios * dias_por_anio;

    %% 3) SERIES DIARIAS
    horas_A3_dia      = datos.horas_A3_dia(:);
    horas_A4_dia      = datos.horas_A4_dia(:);
    horas_A5_dia      = datos.horas_A5_dia(:);
    horas_A8_dia      = datos.horas_A8_dia(:);
    horas_A9_dia      = datos.horas_A9_dia(:);
    horas_horno_dia   = datos.horas_horno_dia(:);
    horas_cargador_dia = datos.horas_cargador_dia(:);

    m_in_horno_kg_dia = datos.masa_entrada_horno_kg_dia(:);
    m_bioc_kg_dia     = datos.masa_bioc_kg_dia(:);
    sacos_40kg_dia    = datos.sacos_40kg_dia(:);

    % Humedad escalar o vector
    hum_in = datos.hum_in_horno_frac;
    if isscalar(hum_in)
        hum_in_horno_frac_dia = hum_in * ones(size(m_in_horno_kg_dia));
    else
        hum_in_horno_frac_dia = hum_in(:);
    end

    n_dias = numel(horas_A3_dia);
    if n_dias ~= dias_total_esp
        error('modelo_energetico_planta: n_dias=%d, esperado=%d.', n_dias,
            dias_total_esp);
    end
end

```

```

    if
any([numel(horas_A4_dia),numel(horas_A5_dia),numel(horas_A8_dia),numel(horas_A9_d
ia), ...
    numel(horas_horno_dia),numel(horas_cargador_dia), ...
    numel(m_in_horno_kg_dia),numel(m_bioc_kg_dia),numel(sacos_40kg_dia),
...
    numel(hum_in_horno_frac_dia)] ~= n_dias)
    error('modelo_energetico_planta: las series diarias no tienen la misma
longitud.');
```

end

% Aux opcionales

```

if isfield(datos,'horas_A12_aux_dia')
    horas_A12_aux_dia = datos.horas_A12_aux_dia(:);
else
    horas_A12_aux_dia = zeros(n_dias,1);
end
if isfield(datos,'horas_admin_dia')
    horas_admin_dia = datos.horas_admin_dia(:);
else
    horas_admin_dia = zeros(n_dias,1);
end
```

%% 4) PARAMETROS ELECTRICOS Y COMBUSTIBLES

```

P_A3_elec_kW = (param.P_A3_kW / max(param.eta_A3,1e-12));
P_A4_elec_kW = (param.P_A4_kW / max(param.eta_A4,1e-12));
P_A5_elec_kW = (param.P_A5_kW / max(param.eta_A5,1e-12));
P_A8_elec_kW = (param.P_A8_kW / max(param.eta_A8,1e-12));
P_A9_elec_kW = (param.P_A9_kW / max(param.eta_A9,1e-12));

P_soplador_elec_kW = (param.P_soplador_horno_kW / max(param.eta_soplador,1e-
12));
P_aux_horno_elec_kW = (param.P_aux_horno_kW /
max(param.eta_aux_horno,1e-12));

P_A12_aux_elec_kW = (param.P_A12_aux_kW / max(param.eta_A12_aux,1e-12));
P_admin_elec_kW = (param.P_admin_kW / max(param.eta_admin, 1e-12));

PCI_GN_kWh_m3 = param.PCI_GN_kWh_m3;
PCI_diesel_kWh_L = param.PCI_diesel_kWh_L;
consumo_diesel_L_h = param.consumo_diesel_L_h;
```

%% 5) ENERGIA ELECTRICA DIARIA

```

E_A3_kWh_dia = P_A3_elec_kW * horas_A3_dia;
E_A4_kWh_dia = P_A4_elec_kW * horas_A4_dia;
E_A5_kWh_dia = P_A5_elec_kW * horas_A5_dia;
E_A8_kWh_dia = P_A8_elec_kW * horas_A8_dia;
E_A9_kWh_dia = P_A9_elec_kW * horas_A9_dia;

```

```

E_soplador_kWh_dia = P_soplador_elec_kW * horas_horno_dia;
E_aux_horno_kWh_dia = P_aux_horno_elec_kW * horas_horno_dia;

```

```

E_A12_aux_kWh_dia = P_A12_aux_elec_kW * horas_A12_aux_dia;
E_admin_kWh_dia = P_admin_elec_kW * horas_admin_dia;

```

```

E_elec_total_kWh_dia = E_A3_kWh_dia + E_A4_kWh_dia + E_A5_kWh_dia + ...
                      E_A8_kWh_dia + E_A9_kWh_dia + ...
                      E_soplador_kWh_dia + E_aux_horno_kWh_dia + ...
                      E_A12_aux_kWh_dia + E_admin_kWh_dia;

```

%% 6) DIESEL DIARIO

```

diesel_L_dia = consumo_diesel_L_h .* horas_cargador_dia;
E_diesel_kWh_dia = diesel_L_dia .* PCI_diesel_kWh_L;

```

%% 7) HORNO GN (balance + estequiometría)

```

[Q_util_kWh_dia, Q_comb_kWh_dia, GN_m3_dia, V_fg_Nm3_dia, m_fg_kg_dia,
desglose_kWh_dia, desglose_MJkg_dia, eta_th_dia, meta_fg] = ...
    modelo_quemador_GN(param, m_in_horno_kg_dia, hum_in_horno_frac_dia,
PCI_GN_kWh_m3);

```

%% 8) ENERGIA TOTAL DIARIA

```

E_total_kWh_dia = E_elec_total_kWh_dia + E_diesel_kWh_dia + Q_comb_kWh_dia;

```

%% 9) AGRUPACION ANUAL

```

E_elec_total_kWh_anual = zeros(n_anios,1);
E_diesel_kWh_anual = zeros(n_anios,1);
E_GN_kWh_anual = zeros(n_anios,1);
GN_m3_anual = zeros(n_anios,1);
Q_util_kWh_anual = zeros(n_anios,1);
E_total_kWh_anual = zeros(n_anios,1);

```

```

bioc_ton_anual      = zeros(n_anios,1);
sacos_anual         = zeros(n_anios,1);
m_in_horno_ton_anual = zeros(n_anios,1);

V_fg_Nm3_anual = zeros(n_anios,1);
m_fg_ton_anual = zeros(n_anios,1);

eta_th_prom_anual   = zeros(n_anios,1);

Qcal_bio_kWh_anual = zeros(n_anios,1);
Qcal_w_kWh_anual   = zeros(n_anios,1);
Qevap_kWh_anual    = zeros(n_anios,1);
Qsup_kWh_anual     = zeros(n_anios,1);
Qtorref_kWh_anual  = zeros(n_anios,1);
Qfg_kWh_anual      = zeros(n_anios,1);
Qperd_kWh_anual    = zeros(n_anios,1);

% --- NUEVO: electricidad anual por equipo ---
E_A3_kWh_anual      = zeros(n_anios,1);
E_A4_kWh_anual      = zeros(n_anios,1);
E_A5_kWh_anual      = zeros(n_anios,1);
E_A8_kWh_anual      = zeros(n_anios,1);
E_A9_kWh_anual      = zeros(n_anios,1);
E_soplador_kWh_anual = zeros(n_anios,1);
E_aux_horno_kWh_anual = zeros(n_anios,1);
E_A12_aux_kWh_anual = zeros(n_anios,1);
E_admin_kWh_anual    = zeros(n_anios,1);

for a = 1:n_anios
    idx = (a-1)*dias_por_anio + (1:dias_por_anio);

    E_elec_total_kWh_anual(a) = sum(E_elec_total_kWh_dia(idx));
    E_diesel_kWh_anual(a)     = sum(E_diesel_kWh_dia(idx));

    E_GN_kWh_anual(a)         = sum(Q_comb_kWh_dia(idx));
    GN_m3_anual(a)            = sum(GN_m3_dia(idx));
    Q_util_kWh_anual(a)       = sum(Q_util_kWh_dia(idx));

```

```

E_total_kWh_anual(a)      = E_elec_total_kWh_anual(a) +
E_diesel_kWh_anual(a) + E_GN_kWh_anual(a);

bioc_ton_anual(a)         = sum(m_bioc_kg_dia(idx)) / 1000;
sacos_anual(a)           = sum(sacos_40kg_dia(idx));
m_in_horno_ton_anual(a)   = sum(m_in_horno_kg_dia(idx)) / 1000;

V_fg_Nm3_anual(a) = sum(V_fg_Nm3_dia(idx));
m_fg_ton_anual(a) = sum(m_fg_kg_dia(idx)) / 1000;

denom = sum(Q_comb_kWh_dia(idx));
if denom > 0
    eta_th_prom_anual(a) = sum(Q_util_kWh_dia(idx)) / denom;
else
    eta_th_prom_anual(a) = 0;
end

Qcal_bio_kWh_anual(a) =
sum(desglose_kWh_dia.Q_calentar_bio_kWh_dia(idx));
Qcal_w_kWh_anual(a)   = sum(desglose_kWh_dia.Q_calentar_w_kWh_dia(idx));
Qevap_kWh_anual(a)    = sum(desglose_kWh_dia.Q_evap_kWh_dia(idx));
Qsup_kWh_anual(a)     = sum(desglose_kWh_dia.Q_superheat_kWh_dia(idx));
Qtorref_kWh_anual(a)  = sum(desglose_kWh_dia.Q_torref_kWh_dia(idx));
Qfg_kWh_anual(a)      = sum(desglose_kWh_dia.Q_fg_kWh_dia(idx));
Qperd_kWh_anual(a)    = sum(desglose_kWh_dia.Q_perd_kWh_dia(idx));

% --- NUEVO: sumas anuales por equipo ---
E_A3_kWh_anual(a)      = sum(E_A3_kWh_dia(idx));
E_A4_kWh_anual(a)      = sum(E_A4_kWh_dia(idx));
E_A5_kWh_anual(a)      = sum(E_A5_kWh_dia(idx));
E_A8_kWh_anual(a)      = sum(E_A8_kWh_dia(idx));
E_A9_kWh_anual(a)      = sum(E_A9_kWh_dia(idx));
E_soplador_kWh_anual(a) = sum(E_soplador_kWh_dia(idx));
E_aux_horno_kWh_anual(a) = sum(E_aux_horno_kWh_dia(idx));
E_A12_aux_kWh_anual(a)  = sum(E_A12_aux_kWh_dia(idx));
E_admin_kWh_anual(a)    = sum(E_admin_kWh_dia(idx));

end

%% 10) AGRUPACION MENSUAL
dias_por_mes = [31 28 31 30 31 30 31 31 30 31 30 31];

```

```

n_meses = 12;

E_elec_mes_kWh = zeros(n_anios, n_meses);
E_diesel_mes_kWh = zeros(n_anios, n_meses);
E_GN_mes_kWh = zeros(n_anios, n_meses);
GN_m3_mes = zeros(n_anios, n_meses);
E_total_mes_kWh = zeros(n_anios, n_meses);

for a = 1:n_anios
    offset = (a-1)*dias_por_anio;
    dia_ini_mes = 1;
    for m = 1:n_meses
        n_dias_mes = dias_por_mes(m);
        idx = offset + (dia_ini_mes : dia_ini_mes + n_dias_mes - 1);

        E_elec_mes_kWh(a,m) = sum(E_elec_total_kWh_dia(idx));
        E_diesel_mes_kWh(a,m) = sum(E_diesel_kWh_dia(idx));
        E_GN_mes_kWh(a,m) = sum(Q_comb_kWh_dia(idx));
        GN_m3_mes(a,m) = sum(GN_m3_dia(idx));

        E_total_mes_kWh(a,m) = E_elec_mes_kWh(a,m) + E_diesel_mes_kWh(a,m) +
E_GN_mes_kWh(a,m);

        dia_ini_mes = dia_ini_mes + n_dias_mes;
    end
end

%% 11) KPI
bioc_ton_safe = max(bioc_ton_anual, 1e-12);
sacos_safe = max(sacos_anual, 1e-12);
m_in_safe = max(m_in_horno_ton_anual, 1e-12);

kWh_ton_elec_bioc = E_elec_total_kWh_anual ./ bioc_ton_safe;
kWh_ton_diesel_bioc = E_diesel_kWh_anual ./ bioc_ton_safe;
kWh_ton_GN_bioc = E_GN_kWh_anual ./ bioc_ton_safe;
kWh_ton_total_bioc = E_total_kWh_anual ./ bioc_ton_safe;

kWh_saco_elec = E_elec_total_kWh_anual ./ sacos_safe;
kWh_saco_diesel = E_diesel_kWh_anual ./ sacos_safe;
kWh_saco_GN = E_GN_kWh_anual ./ sacos_safe;
kWh_saco_total = E_total_kWh_anual ./ sacos_safe;

```

```

Qutil_kWh_por_ton_in = Q_util_kWh_anual ./ m_in_safe;
kWh_GN_por_ton_in    = E_GN_kWh_anual    ./ m_in_safe;
GN_m3_por_ton_in     = GN_m3_anual       ./ m_in_safe;

%% 12) TABLAS
anios_vec = (1:n_anios).';

tabla_elec_por_equipo_anual = table( ...
    anios_vec, ...
    E_A3_kWh_anual, E_A4_kWh_anual, E_A5_kWh_anual, E_A8_kWh_anual,
E_A9_kWh_anual, ...
    E_soplador_kWh_anual, E_aux_horno_kWh_anual, E_A12_aux_kWh_anual,
E_admin_kWh_anual, ...
    E_elec_total_kWh_anual, ...
    'VariableNames', { ...
    'Anio', ...
    'E_A3_kWh', 'E_A4_kWh', 'E_A5_kWh', 'E_A8_kWh', 'E_A9_kWh', ...
    'E_soplador_kWh', 'E_aux_horno_kWh', 'E_A12_aux_kWh', 'E_admin_kWh', ...
    'E_elec_total_kWh'});

tabla_quemador_anual = table( ...
    anios_vec, m_in_horno_ton_anual, ...
    Q_util_kWh_anual, E_GN_kWh_anual, GN_m3_anual, ...
    V_fg_Nm3_anual, m_fg_ton_anual, Qfg_kWh_anual, Qperd_kWh_anual, ...
    eta_th_prom_anual, Qutil_kWh_por_ton_in, kWh_GN_por_ton_in,
GN_m3_por_ton_in, ...
    'VariableNames', { ...
    'Anio', 'Masa_entrada_horno_ton', ...
    'Q_util_kWh_anual', 'E_quim_GN_kWh_anual', 'GN_m3_anual', ...
    'Gases_escape_Nm3_anual', 'Gases_escape_ton_anual', 'Q_fg_kWh_anual', 'Q_perd_kWh_anual' ...
    'eta_th_prom', 'Qutil_kWh_por_ton_in', 'kWh_GN_por_ton_in', 'GN_m3_por_ton_in' });

conv_MJkg = 3.6/1000; % MJ/kg = (kWh/ton)*3.6/1000
tabla_quemador_especifica_MJkg = table( ...
    anios_vec, ...
    Qutil_kWh_por_ton_in*conv_MJkg, ...
    kWh_GN_por_ton_in*conv_MJkg, ...

```



```

(Qfg_kWh_anual./m_in_safe)*conv_MJkg, ...
(Qperd_kWh_anual./m_in_safe)*conv_MJkg, ...
eta_th_prom_anual, ...
(Qcal_bio_kWh_anual./m_in_safe)*conv_MJkg, ...
(Qcal_w_kWh_anual./m_in_safe)*conv_MJkg, ...
(Qevap_kWh_anual./m_in_safe)*conv_MJkg, ...
(Qsup_kWh_anual./m_in_safe)*conv_MJkg, ...
(Qtorref_kWh_anual./m_in_safe)*conv_MJkg, ...
'VariableNames', { ...
'Anio', ...

'Qutil_MJkg_in', 'E_quim_GN_MJkg_in', 'Q_fg_MJkg_in', 'Q_perd_MJkg_in', 'eta_th_prom'
, ...

'MJkg_calentar_bio', 'MJkg_calentar_agua', 'MJkg_evap', 'MJkg_superheat', 'MJkg_torre
f'}});

tabla_energia_anual = table( ...
    anios_vec, ...
    E_elec_total_kWh_anual, E_diesel_kWh_anual, E_GN_kWh_anual,
E_total_kWh_anual, ...
    'VariableNames',
{'Anio', 'E_elec_kWh', 'E_diesel_kWh', 'E_GN_kWh', 'E_total_kWh'});

tabla_kpi_kWh_ton_bioc = table( ...
    anios_vec, bioc_ton_anual, ...
    kWh_ton_elec_bioc, kWh_ton_diesel_bioc, kWh_ton_GN_bioc,
kWh_ton_total_bioc, ...
    'VariableNames',
{'Anio', 'Biocarbon_ton', 'kWh_ton_elec', 'kWh_ton_diesel', 'kWh_ton_GN', 'kWh_ton_tot
al'});

tabla_kpi_kWh_saco = table( ...
    anios_vec, sacos_anual, ...
    kWh_saco_elec, kWh_saco_diesel, kWh_saco_GN, kWh_saco_total, ...
    'VariableNames',
{'Anio', 'Sacos_40kg', 'kWh_saco_elec', 'kWh_saco_diesel', 'kWh_saco_GN', 'kWh_saco_to
tal'});

%% 13) SALIDA
res_ener = struct();

res_ener.E_elec_total_kWh_dia = E_elec_total_kWh_dia;

```

```

res_ener.E_diesel_kWh_dia      = E_diesel_kWh_dia;
res_ener.E_GN_kWh_dia         = Q_comb_kWh_dia;
res_ener.GN_m3_dia            = GN_m3_dia;
res_ener.Q_util_kWh_dia       = Q_util_kWh_dia;
res_ener.E_total_kWh_dia      = E_total_kWh_dia;

```

```

res_ener.E_elec_total_kWh_anual = E_elec_total_kWh_anual;
res_ener.E_diesel_kWh_anual     = E_diesel_kWh_anual;
res_ener.E_GN_kWh_anual         = E_GN_kWh_anual;
res_ener.GN_m3_anual            = GN_m3_anual;
res_ener.Q_util_kWh_anual       = Q_util_kWh_anual;
res_ener.E_total_kWh_anual      = E_total_kWh_anual;

```

```

res_ener.V_fg_Nm3_dia  = V_fg_Nm3_dia;
res_ener.m_fg_kg_dia   = m_fg_kg_dia;
res_ener.V_fg_Nm3_anual = V_fg_Nm3_anual;
res_ener.m_fg_ton_anual = m_fg_ton_anual;

```

```

res_ener.E_elec_mes_kWh      = E_elec_mes_kWh;
res_ener.E_diesel_mes_kWh    = E_diesel_mes_kWh;
res_ener.E_GN_mes_kWh        = E_GN_mes_kWh;
res_ener.GN_m3_mes           = GN_m3_mes;
res_ener.E_total_mes_kWh     = E_total_mes_kWh;

```

```

res_ener.tabla_elec_por_equipo_anual = tabla_elec_por_equipo_anual;

```

% (opcional, por si los quieres luego)

```

res_ener.E_A3_kWh_anual = E_A3_kWh_anual;
res_ener.E_A4_kWh_anual = E_A4_kWh_anual;
res_ener.E_A5_kWh_anual = E_A5_kWh_anual;
res_ener.E_A8_kWh_anual = E_A8_kWh_anual;
res_ener.E_A9_kWh_anual = E_A9_kWh_anual;
res_ener.E_soplador_kWh_anual = E_soplador_kWh_anual;
res_ener.E_aux_horno_kWh_anual = E_aux_horno_kWh_anual;
res_ener.E_A12_aux_kWh_anual = E_A12_aux_kWh_anual;
res_ener.E_admin_kWh_anual = E_admin_kWh_anual;

```

```

res_ener.tabla_quemador_anual      = tabla_quemador_anual;
res_ener.tabla_quemador_especifica_MJkg = tabla_quemador_especifica_MJkg;
res_ener.tabla_energia_anual       = tabla_energia_anual;

```

```

res_ener.tabla_kpi_kWh_ton_bioc      = tabla_kpi_kWh_ton_bioc;
res_ener.tabla_kpi_kWh_saco_40kg     = tabla_kpi_kWh_saco;

res_ener.desglose_quemador_kWh_dia = desglose_kWh_dia;
res_ener.desglose_quemador_MJkg_dia = desglose_MJkg_dia;
res_ener.eta_th_dia = eta_th_dia;

res_ener.meta_fg = meta_fg;
% Accesos directos (comodidad)
res_ener.V_fg_Nm3_por_Nm3_GN = meta_fg.V_fg_Nm3_por_Nm3_GN;
res_ener.m_fg_kg_por_Nm3_GN   = meta_fg.m_fg_kg_por_Nm3_GN;
res_ener.cp_fg_J_kgK          = meta_fg.cp_fg_J_kgK;

%% 14) RESUMEN
fprintf('--- MODELO ENERGETICO PLANTA (con GN) ---\n');
fprintf('Energia electrica total %d anios: %.0f kWh\n', n_anios,
sum(E_elec_total_kWh_anual));
fprintf('Energia diesel total %d anios   : %.0f kWh\n', n_anios,
sum(E_diesel_kWh_anual));
fprintf('Energia GN total %d anios         : %.0f kWh\n', n_anios,
sum(E_GN_kWh_anual));
fprintf('Consumo GN total %d anios          : %.0f m3\n', n_anios,
sum(GN_m3_anual));
fprintf('Energia total %d anios              : %.0f kWh\n', n_anios,
sum(E_total_kWh_anual));
fprintf('-----\n');
end

% =====
% SUBFUNCIÓN: QUEMADOR GN (balance con gases + pérdidas + estequiometría)
% =====
function [Q_req_kWh_dia, Q_comb_kWh_dia, GN_m3_dia, V_fg_Nm3_dia, m_fg_kg_dia,
desglose_kWh_dia, desglose_MJkg_dia, eta_th , meta_fg] = ...
    modelo_quemador_GN(param, m_wet_kg_dia, hum_frac_dia, PCI_GN_kWh_m3)

hum    = max(min(hum_frac_dia, 0.95), 0);
m_wet_kg_dia = max(m_wet_kg_dia, 0);

T_in    = param.T_in_C;
T_proc  = param.T_proc_C;
T_boil  = 100;

```

```

T_amb = param.T_amb_C;
T_fg   = param.T_fg_C;

dT_bio = max(T_proc - T_in, 0);
dT_wliq = max(T_boil - T_in, 0);
dT_wvap = max(T_proc - T_boil, 0);
dT_fg   = max(T_fg - T_amb, 0);

cp_bio = param.cp_bio_J_kgK;
cp_wl  = param.cp_w_liq_J_kgK;
cp_wv  = param.cp_w_vap_J_kgK;
h_vap  = param.h_vap_J_kg;
Q_torr = param.Q_torref_J_kg_dry;

m_w = hum      .* m_wet_kg_dia;
m_d = (1 - hum) .* m_wet_kg_dia;

Q_calentar_bio_J = m_d .* cp_bio .* dT_bio;
Q_calentar_w_J   = m_w .* cp_wl  .* dT_wliq;
Q_evap_J         = m_w .* h_vap;
Q_superheat_J    = m_w .* cp_wv  .* dT_wvap;
Q_torref_J       = m_d .* Q_torr;

Q_req_J_dia = Q_calentar_bio_J + Q_calentar_w_J + Q_evap_J + Q_superheat_J
+ Q_torref_J;
Q_req_kWh_dia = Q_req_J_dia / 3.6e6;

lambda = max(param.lambda_air, 1.0);

n_CO2 = 1;
n_H2O = 2;
n_O2  = 2*lambda - 2;
n_N2  = 7.52*lambda;
n_tot = n_CO2 + n_H2O + n_O2 + n_N2;

x_CO2 = n_CO2 / n_tot;
x_H2O = n_H2O / n_tot;
x_O2  = n_O2  / n_tot;
x_N2  = n_N2  / n_tot;

```

```

cp_mix_kJ_kmolK = x_CO2*param.cp_CO2_kJ_kmolK + x_H2O*param.cp_H2O_kJ_kmolK +
...
x_O2*param.cp_O2_kJ_kmolK + x_N2*param.cp_N2_kJ_kmolK;

M_mix = x_CO2*param.M_CO2 + x_H2O*param.M_H2O + x_O2*param.M_O2 +
x_N2*param.M_N2;

cp_fg_J_kgK = (cp_mix_kJ_kmolK * 1e3) / max(M_mix, 1e-12);

rho_fg_kg_Nm3 = M_mix / 22.414;
m_fg_kg_por_Nm3_GN = n_tot * rho_fg_kg_Nm3;

% : volumen de gases por Nm3 de GN (a CN)
V_fg_Nm3_por_Nm3_GN = n_tot; % 1 Nm3 CH4 -> n_tot Nm3 gases (misma base CN)

f = max(min(param.frac_perdidas, 0.50), 0);

A = PCI_GN_kWh_m3 * 3.6e6;
B = m_fg_kg_por_Nm3_GN * cp_fg_J_kgK * dT_fg;

denom = A*(1 - f) - B;
if denom <= 0
    error('modelo_quemador_GN: Denominador <= 0. Revisa T_fg, lambda_air,
frac_perdidas, cp o PCI.');
```

```

end

GN_m3_dia = Q_req_J_dia ./ denom;

% ---: flujos de gases de escape (base CN) ---
V_fg_Nm3_dia = GN_m3_dia .* V_fg_Nm3_por_Nm3_GN; % [Nm3/día]
m_fg_kg_dia = GN_m3_dia .* m_fg_kg_por_Nm3_GN; % [kg/día]
Q_comb_J_dia = GN_m3_dia .* A;
Q_fg_J_dia = GN_m3_dia .* B;
Q_perd_J_dia = f .* Q_comb_J_dia;

Q_comb_kWh_dia = Q_comb_J_dia / 3.6e6;
Q_fg_kWh_dia = Q_fg_J_dia / 3.6e6;
Q_perd_kWh_dia = Q_perd_J_dia / 3.6e6;

```

```

eta_th = Q_req_J_dia ./ max(Q_comb_J_dia, 1e-12);

desglose_kWh_dia = struct();
desglose_kWh_dia.Q_calentar_bio_kWh_dia = Q_calentar_bio_J / 3.6e6;
desglose_kWh_dia.Q_calentar_w_kWh_dia   = Q_calentar_w_J   / 3.6e6;
desglose_kWh_dia.Q_evap_kWh_dia         = Q_evap_J         / 3.6e6;
desglose_kWh_dia.Q_superheat_kWh_dia    = Q_superheat_J    / 3.6e6;
desglose_kWh_dia.Q_torref_kWh_dia       = Q_torref_J       / 3.6e6;

desglose_kWh_dia.Q_util_kWh_dia          = Q_req_kWh_dia;
desglose_kWh_dia.Q_fg_kWh_dia            = Q_fg_kWh_dia;
desglose_kWh_dia.Q_perd_kWh_dia          = Q_perd_kWh_dia;
desglose_kWh_dia.Q_comb_kWh_dia          = Q_comb_kWh_dia;
desglose_kWh_dia.eta_th_dia              = eta_th;

m_total = max(m_wet_kg_dia, 1e-9);
desglose_MJkg_dia = struct();
desglose_MJkg_dia.Qutil_MJkg = (Q_req_J_dia / 1e6) ./ m_total;
desglose_MJkg_dia.Qcomb_MJkg = (Q_comb_J_dia / 1e6) ./ m_total;
desglose_MJkg_dia.Qfg_MJkg   = (Q_fg_J_dia / 1e6) ./ m_total;
desglose_MJkg_dia.Qperd_MJkg = (Q_perd_J_dia / 1e6) ./ m_total;

meta_fg = struct();
meta_fg.V_fg_Nm3_por_Nm3_GN = n_tot; % 1 Nm3 CH4 -> n_tot Nm3 gases
(CN)
meta_fg.m_fg_kg_por_Nm3_GN   = m_fg_kg_por_Nm3_GN; % kg gases / Nm3 GN
meta_fg.cp_fg_J_kgK          = cp_fg_J_kgK;        % cp mezcla (base masa)
meta_fg.M_mix_kg_kmol        = M_mix;              % opcional

end

```

```

function res_ex = modelo_exergetico_planta(param_ex, param, res_ener, datos_ex)
% MODELO_EXERGETICO_PLANTA
% -----
% Submodelo exergético independiente del energético.
% HORNO (A10) a base anual.
% ENTRADAS:
%   param_ex : parámetros SOLO exergía (beta, LHV, cp productos, etc.)
%   param     : parámetros globales (n_anios, dias_por_anio, T_amb_C, etc.)
%   res_ener  : salida del modelo energético (GN_m3_anual, meta_fg, etc.)
%   datos_ex  : struct con series diarias del horno (desde el script principal)
%       .m_in_horno_kg_dia
%       .m_BC_kg_dia, .m_liq_kg_dia, m_gas_kg_dia
%       .E_el_horno_kWh_dia (soplador+aux en un solo vector)
%       .hum_in_horno_frac (escalar o vector) -> para pasar a base seca
%
% SALIDAS:
%   res_ex : struct con tablas/anuales del horno (Ex_D y eta_ex)
% -----

%% 0) Validaciones mínimas
req =
{'m_in_horno_kg_dia', 'm_BC_kg_dia', 'm_liq_kg_dia', 'm_gas_kg_dia', 'E_el_horno_kWh_dia',
'hum_in_horno_frac'};
for i = 1:numel(req)
    if ~isfield(datos_ex, req{i})
        error('modelo_exergetico_planta: falta datos_ex.%s', req{i});
    end
end
if ~isfield(res_ener, 'GN_m3_anual')
    error('modelo_exergetico_planta: res_ener no tiene GN_m3_anual');
end
if ~isfield(res_ener, 'meta_fg')
    error('modelo_exergetico_planta: res_ener.meta_fg no existe (agrega el cambio
en energético).');
end

n_anios      = param.n_anios;
dias_por_anio = param.dias_por_anio;
n_dias_esp    = n_anios * dias_por_anio;

% Asegurar columnas
m_in_horno = datos_ex.m_in_horno_kg_dia(:);
m_BC       = datos_ex.m_BC_kg_dia(:);
m_liq      = datos_ex.m_liq_kg_dia(:);
m_gas      = datos_ex.m_gas_kg_dia(:);
E_el_horno_kWh_dia = datos_ex.E_el_horno_kWh_dia(:);

if
any([numel(m_in_horno), numel(m_BC), numel(m_liq), numel(m_gas), numel(E_el_horno_kWh_dia)
]) ~= n_dias_esp)
    error('modelo_exergetico_planta: series diarias no coinciden con
n_anios*dias_por_anio.');
```

```

end

% Humedad (escalar o vector)
```

```

hum_in = datos_ex.hum_in_horno_frac;
if isscalar(hum_in)
    hum_dia = hum_in * ones(n_dias_esp,1);
else
    hum_dia = hum_in(:);
    if numel(hum_dia) ~= n_dias_esp
        error('modelo_exergetico_planta: hum_in_horno_frac debe ser escalar o
vector n_dias.');
```

end

```

end

%% 1) Agregación anual de variables base (kg/año, kWh/año)
m_in_horno_kg_anual = zeros(n_anios,1);
m_bio_dry_kg_anual = zeros(n_anios,1);
m_BC_kg_anual = zeros(n_anios,1);
m_liq_kg_anual = zeros(n_anios,1);
m_gas_kg_anual = zeros(n_anios,1);
E_el_horno_kWh_anual = zeros(n_anios,1);

for a = 1:n_anios
    idx = (a-1)*dias_por_anio + (1:dias_por_anio);

    m_in_horno_kg_anual(a) = sum(m_in_horno(idx));
    m_BC_kg_anual(a) = sum(m_BC(idx));
    m_liq_kg_anual(a) = sum(m_liq(idx));
    m_gas_kg_anual(a) = sum(m_gas(idx));

    E_el_horno_kWh_anual(a) = sum(E_el_horno_kWh_dia(idx));

    % Biomasa seca anual (a partir de humedad de alimentación)
    % m_dry = (1 - hum) * m_wet
    m_bio_dry_kg_anual(a) = sum( (1 - max(min(hum_dia(idx),0.95),0)) .*
m_in_horno(idx) );
end

%% 2) Preparar "param_ex" a partir del energético (gases de combustión)
% Temperaturas base (K)
if ~isfield(param_ex,'T0_K')
    % Si no lo defines, usamos ambiente del param energético (°C -> K)
    param_ex.T0_K = (param.T_amb_C + 273.15);
end
if ~isfield(param_ex,'T_fg_K')
    param_ex.T_fg_K = (param.T_fg_C + 273.15);
end
if ~isfield(param_ex,'T_proc_K')
    param_ex.T_proc_K = (param.T_proc_C + 273.15);
end

% "meta_fg" viene del energético
param_ex.cp_fg_J_kgK = res_ener.meta_fg.cp_fg_J_kgK;
param_ex.m_fg_kg_por_Nm3_GN = res_ener.meta_fg.m_fg_kg_por_Nm3_GN;
param_ex.V_fg_Nm3_por_Nm3_GN = res_ener.meta_fg.V_fg_Nm3_por_Nm3_GN;

%% 3) Armar "datos_horno" (anual) para el horno
datos_horno = struct();

```



```

datos_horno.GN_m3_anual      = res_ener.GN_m3_anual(:);

% Electricidad del horno anual (kWh): se convierte a MJ dentro
datos_horno.E_el_horno_kWh_anual = E_el_horno_kWh_anual;

% Biomasa seca anual (kg/año)
datos_horno.m_bio_dry_kg_anual  = m_bio_dry_kg_anual;

% Productos pirólisis (kg/año)
datos_horno.m_BC_kg_anual      = m_BC_kg_anual;
datos_horno.m_liq_kg_anual     = m_liq_kg_anual;
datos_horno.m_gas_kg_anual     = m_gas_kg_anual;

% Propiedades químicas (MJ/kg, MJ/Nm3) y betas (adimensional)
% Nota: param_ex debe traer estos campos (ver bloque de parámetros en el
principal)
datos_horno.PCI_GN_MJ_m3      = param_ex.PCI_GN_MJ_m3;
datos_horno.beta_GN          = param_ex.beta_GN;

datos_horno.LHV_bio_MJ_kg     = param_ex.LHV_bio_MJ_kg;
datos_horno.beta_bio         = param_ex.beta_bio;

datos_horno.LHV_BC_MJ_kg      = param_ex.LHV_BC_MJ_kg;
datos_horno.beta_BC          = param_ex.beta_BC;

% Conversión masa → volumen gas pirólisis (CN)
datos_horno.V_gas_piro_Nm3_anual = ...
    datos_horno.m_gas_kg_anual ./ param_ex.rho_gas_piro_kg_Nm3;

%% 4) Calcular exergía horno (función interna)
out_horno = modelo_exergetico_horno(param_ex, datos_horno);

%% 5) Tabla anual de resultados (horno) - con exergías principales para ver
magnitudes
anios_vec = (1:n_anios).';

tabla_ex_horno_anual = table( anios_vec, out_horno.Ex_in_MJ_anual,
    out_horno.Ex_out_MJ_anual, out_horno.Ex_D_horno_MJ_anual,...
    out_horno.Ex_ch_bio_MJ_anual, out_horno.Ex_ch_GN_MJ_anual,
    out_horno.Ex_ch_BC_MJ_anual , out_horno.Ex_ch_liq_MJ_anual,
    out_horno.Ex_ch_gas_MJ_anual, ...
    out_horno.Ex_el_horno_MJ_anual , out_horno.Ex_ph_fg_MJ_anual,
    out_horno.Ex_ph_prod_MJ_anual, out_horno.eta_ex_horno, ...
    'VariableNames', { 'Anio','Ex_in_MJ_anual','Ex_out_MJ_anual',
    'Ex_D_horno_MJ_anual','Ex_ch_bio_MJ_anual','Ex_ch_GN_MJ_anual', 'Ex_ch_BC_MJ_anual',
    ...
    'Ex_ch_liq_MJ_anual','Ex_ch_gas_MJ_anual','Ex_el_horno_MJ_anual',
    'Ex_ph_fg_MJ_anual', 'Ex_ph_prod_MJ_anual','eta_ex_horno' } );

% ===== Porcentajes exergeticos representativos (un solo año) =====
k = 1; % año representativo (1 o 15, da igual)

Ex_in = out_horno.Ex_in_MJ_anual(k);

pct_bio = 100 * out_horno.Ex_ch_bio_MJ_anual(k) / Ex_in;

```

```

pct_GN    = 100 * out_horno.Ex_ch_GN_MJ_anual(k) / Ex_in;
pct_el    = 100 * out_horno.Ex_el_horno_MJ_anual(k) / Ex_in;
pct_BC    = 100 * out_horno.Ex_ch_BC_MJ_anual(k) / Ex_in;
p_liq    = 100*out_horno.Ex_ch_liq_MJ_anual(k)/Ex_in;
p_gas    = 100*out_horno.Ex_ch_gas_MJ_anual(k)/Ex_in;
pct_D     = 100 * out_horno.Ex_D_horno_MJ_anual(k) / Ex_in;

fprintf('\n--- Horno de pirolisis: participacion exergetica (anio %d) ---\n', k);
fprintf('Exergia biomasa      : %6.2f %%\n', pct_bio);
fprintf('Exergia gas natural    : %6.2f %%\n', pct_GN);
fprintf('Exergia electrica      : %6.2f %%\n', pct_el);
fprintf('Exergia biocarbon      : %6.2f %% (util)\n', pct_BC);
fprintf('Exergia liquidos       : %6.2f %%\n', p_liq);
fprintf('Exergia gas piro       : %6.2f %%\n', p_gas);
fprintf('Exergia destruida      : %6.2f %%\n', pct_D);
fprintf('-----\n');

mref = datos_horno.m_bio_dry_kg_anual(:);    % [kg/año] base normalización
mref = max(mref, 1e-12);

tabla_ex_horno_MJkg_in = table( ...
    anios_vec, ...
    out_horno.Ex_in_MJ_anual./mref, ...
    out_horno.Ex_out_MJ_anual./mref, ...
    out_horno.Ex_D_horno_MJ_anual./mref, ...
    out_horno.Ex_ch_bio_MJ_anual./mref, ...
    out_horno.Ex_ch_GN_MJ_anual./mref, ...
    out_horno.Ex_ch_BC_MJ_anual./mref, ...
    out_horno.Ex_ch_liq_MJ_anual./mref, ...
    out_horno.Ex_ch_gas_MJ_anual./mref, ...
    out_horno.Ex_el_horno_MJ_anual./mref, ...
    out_horno.Ex_ph_fg_MJ_anual./mref, ...
    out_horno.Ex_ph_prod_MJ_anual./mref, ...
    out_horno.eta_ex_horno, ...
    'VariableNames', { ...
    'Anio', ...
    'Ex_in_MJkg_in', 'Ex_out_MJkg_in', 'Ex_D_horno_MJkg_in', ...
    'Ex_ch_bio_MJkg_in', 'Ex_ch_GN_MJkg_in', 'Ex_ch_BC_MJkg_in', ...
    'Ex_ch_liq_MJkg_in', 'Ex_ch_gas_MJkg_in', 'Ex_el_horno_MJkg_in', ...
    'Ex_ph_fg_MJkg_in', 'Ex_ph_prod_MJkg_in', 'eta_ex_horno'} );

%% 6) Empaquetar salida
res_ex = struct();
res_ex.tabla_ex_horno_anual    = tabla_ex_horno_anual;
res_ex.tabla_ex_horno_MJkg_in = tabla_ex_horno_MJkg_in;
res_ex.out_horno              = out_horno;

end

% =====
% FUNCIÓN INTERNA: HORNO (BASE ANUAL, MJ)
% =====

```

```

function out_ex = modelo_exergetico_horno(param_ex, datos)

    eps0 = 1e-12;

    % Asegurar columna
    GN_m3_anual      = datos.GN_m3_anual(:);
    m_bio_dry_kg_anual = datos.m_bio_dry_kg_anual(:);
    m_BC_kg_anual     = datos.m_BC_kg_anual(:);
    m_liq_kg_anual     = datos.m_liq_kg_anual(:);
    m_gas_kg_anual     = datos.m_gas_kg_anual(:);

    % Electricidad: se recibe kWh/año -> convertir a MJ/año
    E_el_horno_MJ_anual = 3.6 * datos.E_el_horno_kWh_anual(:);

    % ----- 1) ENTRADAS -----
    ex_ch_GN_MJ_m3      = datos.beta_GN .* datos.PCI_GN_MJ_m3;
    Ex_ch_GN_MJ_anual   = GN_m3_anual .* ex_ch_GN_MJ_m3;

    Ex_el_horno_MJ_anual = E_el_horno_MJ_anual;

    ex_ch_bio_MJ_kg      = datos.beta_bio .* datos.LHV_bio_MJ_kg;
    Ex_ch_bio_MJ_anual   = m_bio_dry_kg_anual .* ex_ch_bio_MJ_kg;

    % ----- 2) SALIDAS -----
    ex_ch_BC_MJ_kg      = datos.beta_BC .* datos.LHV_BC_MJ_kg;
    Ex_ch_BC_MJ_anual   = m_BC_kg_anual .* ex_ch_BC_MJ_kg;

    % Líquidos de pirólisis: ex_ch = beta_liq * LHV_liq [MJ/kg]
    ex_ch_liq_MJ_kg      = param_ex.beta_liq .* param_ex.LHV_liq_MJ_kg;
    Ex_ch_liq_MJ_anual   = m_liq_kg_anual .* ex_ch_liq_MJ_kg;

    % 2.2 Gas de pirólisis: ex_ch = beta_gas * LHV_gas [MJ/Nm3]
    V_gas_piro_Nm3_anual = datos.V_gas_piro_Nm3_anual(:);
    ex_ch_gas_MJ_Nm3      = param_ex.beta_gas .* param_ex.LHV_gas_MJ_Nm3;
    Ex_ch_gas_MJ_anual    = V_gas_piro_Nm3_anual .* ex_ch_gas_MJ_Nm3;

    % Gases de combustión: masa y exergía física
    m_fg_kg_anual = GN_m3_anual .* param_ex.m_fg_kg_por_Nm3_GN;

    T0 = param_ex.T0_K;
    Tfg = param_ex.T_fg_K;

    delta_T_fg = Tfg - T0;
    log_T_fg_T0 = log(Tfg ./ T0);

    Ex_ph_fg_J_anual = m_fg_kg_anual .* param_ex.cp_fg_J_kgK .* (delta_T_fg - T0 .*
log_T_fg_T0);
    Ex_ph_fg_MJ_anual = Ex_ph_fg_J_anual ./ 1e6;

    % Productos a T_proc
    Tp = param_ex.T_proc_K;
    delta_T_proc = Tp - T0;
    log_T_proc_T0 = log(Tp ./ T0);

```

```

mcp_prod_J_K_anual = ...
    m_BC_kg_anual .* param_ex.cp_BC_J_kgK + ...
    m_liq_kg_anual .* param_ex.cp_liq_J_kgK + ...
    m_gas_kg_anual .* param_ex.cp_gas_J_kgK;

Ex_ph_prod_J_anual = mcp_prod_J_K_anual .* (delta_T_proc - T0 .* log_T_proc_T0);
Ex_ph_prod_MJ_anual = Ex_ph_prod_J_anual ./ 1e6;

% ----- 3) BALANCE EXERGÉTICO -----
Ex_in_MJ_anual = Ex_ch_GN_MJ_anual + Ex_el_horno_MJ_anual + Ex_ch_bio_MJ_anual;
Ex_out_MJ_anual = Ex_ch_BC_MJ_anual + Ex_ch_liq_MJ_anual + Ex_ch_gas_MJ_anual +
Ex_ph_fg_MJ_anual + Ex_ph_prod_MJ_anual ;

% Exergía destruida
Ex_D_horno_MJ_anual = Ex_in_MJ_anual - Ex_out_MJ_anual;

% ----- 4) EFICIENCIA EXERGÉTICA (clásica, <=1) -----
% Útil: exergía química del biocarbón producido
Ex_util_MJ_anual = Ex_ch_BC_MJ_anual + Ex_ch_liq_MJ_anual + Ex_ch_gas_MJ_anual;

% Insumos: exergía total que entra (biomasa + GN + electricidad)
eta_ex_horno = Ex_util_MJ_anual ./ max(Ex_in_MJ_anual, 1e-12);

% ----- 5) SALIDA -----
out_ex = struct();

out_ex.Ex_ch_GN_MJ_anual = Ex_ch_GN_MJ_anual;
out_ex.Ex_el_horno_MJ_anual = Ex_el_horno_MJ_anual;
out_ex.Ex_ch_bio_MJ_anual = Ex_ch_bio_MJ_anual;

out_ex.Ex_ch_BC_MJ_anual = Ex_ch_BC_MJ_anual;
out_ex.Ex_ph_fg_MJ_anual = Ex_ph_fg_MJ_anual;
out_ex.Ex_ph_prod_MJ_anual = Ex_ph_prod_MJ_anual;

out_ex.Ex_in_MJ_anual = Ex_in_MJ_anual;
out_ex.Ex_out_MJ_anual = Ex_out_MJ_anual;

out_ex.Ex_D_horno_MJ_anual = Ex_D_horno_MJ_anual;
out_ex.eta_ex_horno = eta_ex_horno;

out_ex.Ex_ch_liq_MJ_anual = Ex_ch_liq_MJ_anual;
out_ex.Ex_ch_gas_MJ_anual = Ex_ch_gas_MJ_anual;
out_ex.Ex_util_MJ_anual = Ex_util_MJ_anual;

out_ex.V_gas_piro_Nm3_anual = V_gas_piro_Nm3_anual;

% Extras útiles
out_ex.m_fg_kg_anual = m_fg_kg_anual;
out_ex.V_fg_Nm3_anual = GN_m3_anual .* param_ex.V_fg_Nm3_por_Nm3_GN;

out_ex.GN_m3_anual = GN_m3_anual;

end

```

```

function res_econ = modelo_economico_planta(param_econ, res_ener,
tabla_masas_anual)
% MODELO_ECONOMICO_PLANTA
% -----
% Calcula ingresos, OPEX (energía + fijo), CAPEX, flujo de caja y métricas:
% VPN, TIR (sin toolbox), Payback, precio mínimo del bulto.
%
% ENTRADAS:
%   param_econ      : struct parámetros económicos (USD)
%   res_ener        : struct del modelo energético (anual)
%   tabla_masas_anual : table con columnas:
%
%   'Anio', 'Masa_recibida_ton', 'Biocarbon_ton', 'Lixiviado_real_L', 'Liquidos_piro_L'
%
% SALIDAS:
%   res_econ : struct con tablas y métricas
% -----
%
%   %% ----- Validaciones mínimas -----
%   req_cols =
{'Anio', 'Masa_recibida_ton', 'Biocarbon_ton', 'Lixiviado_real_L', 'Liquidos_piro_L'}
;
    for i = 1:numel(req_cols)
        if ~ismember(req_cols{i}, tabla_masas_anual.Properties.VariableNames)
            error('modelo_economico_planta: tabla_masas_anual no tiene columna
%s', req_cols{i});
        end
    end
    n = param_econ.n_anios;
    if height(tabla_masas_anual) ~= n
        error('modelo_economico_planta: tabla_masas_anual tiene %d filas,
esperado %d', height(tabla_masas_anual), n);
    end
    if ~isfield(res_ener, 'E_elec_total_kWh_anual')
        error('modelo_economico_planta: res_ener no tiene
E_elec_total_kWh_anual');
    end
    if ~isfield(res_ener, 'E_diesel_kWh_anual')
        error('modelo_economico_planta: res_ener no tiene E_diesel_kWh_anual');
    end
    if ~isfield(res_ener, 'GN_m3_anual')
        error('modelo_economico_planta: res_ener no tiene GN_m3_anual');
    end
    if ~isfield(param_econ, 'PCI_diesel_kWh_L')
        error('modelo_economico_planta: falta param_econ.PCI_diesel_kWh_L');
    end
    % OPEX mantto: en tu planteamiento lo estás usando fijo (5310 USD/año)
    if ~isfield(param_econ, 'mantto_USD_anual')

```

```

        error('modelo_economico_planta: falta param_econ.mantto_USD_anual (ej:
5310)');
    end
    r = param_econ.tasa_descuento;
    %% ----- Producción anual -----
    anio = tabla_masas_anual.Anio(:); % 1..n
    masa_rec_ton = tabla_masas_anual.Masa_recibida_ton(:);
    bioc_ton = tabla_masas_anual.Biocarbon_ton(:);
    lix_real_L = tabla_masas_anual.Lixiviado_real_L(:);
    liq_piro_L = tabla_masas_anual.Liquidos_piro_L(:);
    liq_total_L = lix_real_L + liq_piro_L;
    % >>> Define qué líquido se monetiza <<<
    % Recomendación: monetizar SOLO lixiviado real (drenajes), NO líquidos de
    pirólisis.
    lix_monetizable_L = lix_real_L;
    % Bultos 40 kg por año
    bultos_40kg = (bioc_ton * 1000) / 40;
    %% ----- Precios con crecimiento -----
    p_bulto = param_econ.precio_bulto_USD * (1 + param_econ.g_bulto).^(anio-
1);
    p_ahorro = param_econ.ahorro_disp_USD_ton * (1 + param_econ.g_ahorro).^(anio-
1);
    p_lix = param_econ.precio_lix_USD_L * (1 + param_econ.g_lix).^(anio-
1);
    p_elec = param_econ.p_elec_USD_kWh_1 * (1 + param_econ.g_elec).^(anio-1);
    p_diesel = param_econ.p_diesel_USD_L_1 * (1 + param_econ.g_diesel).^(anio-1);
    p_gn = param_econ.p_gn_USD_m3_1 * (1 + param_econ.g_gn).^(anio-1);
    salario_USD_mes_anual = param_econ.salario_USD_mes * (1 +
param_econ.g_salario).^(anio-1);

    %% ----- Energía anual -----
    E_elec_kWh_anual = res_ener.E_elec_total_kWh_anual(:);
    E_diesel_kWh_anual = res_ener.E_diesel_kWh_anual(:);
    GN_m3_anual = res_ener.GN_m3_anual(:);
    if numel(E_elec_kWh_anual) ~= n || numel(E_diesel_kWh_anual) ~= n ||
numel(GN_m3_anual) ~= n
        error('modelo_economico_planta: tamaños de energía anual no coinciden con
n_anios');
    end
    diesel_L_anual = E_diesel_kWh_anual / param_econ.PCI_diesel_kWh_L;
    %% ----- Ingresos -----
    Ing_bioc_USD = bultos_40kg .* p_bulto;
    Ing_ahorro_USD = masa_rec_ton .* p_ahorro;
    Ing_lix_USD = lix_monetizable_L .* p_lix;
    Ing_total_USD = Ing_bioc_USD + Ing_ahorro_USD + Ing_lix_USD;
    %% ----- Costos energía -----
    C_elec_USD = E_elec_kWh_anual .* p_elec;

```

```

C_diesel_USD = diesel_L_anual .* p_diesel;
C_gn_USD      = GN_m3_anual      .* p_gn;
OPEX_energia_USD = C_elec_USD + C_diesel_USD + C_gn_USD;
%% ----- OPEX fijo anual -----
OPEX_manoobra_USD = param_econ.n_trabajadores * 12 .* salario_USD_mes_anual;

% Mantto fijo anual (tu valor)
OPEX_mantto_USD = param_econ.mantto_USD_anual * ones(n,1);
% Insumos (2% CAPEX)
OPEX_insumos_USD = (param_econ.frac_insumos * param_econ.capex_USD) *
ones(n,1);
OPEX_base_USD = OPEX_energia_USD + OPEX_manoobra_USD + OPEX_mantto_USD +
OPEX_insumos_USD;
% Otros = 2% del OPEX total => OPEX_total = base / (1 - frac_otros)
OPEX_total_USD = OPEX_base_USD / (1 - param_econ.frac_otros);
OPEX_otros_USD = OPEX_total_USD - OPEX_base_USD;
%% ----- Flujo de caja -----
CF_anual_USD = Ing_total_USD - OPEX_total_USD;
CF = zeros(n+1,1);
CF(1) = -param_econ.capex_USD;    % Año 0
CF(2:end) = CF_anual_USD;        % Años 1..n
%% ----- VPN -----
t = (0:n)';
DF = 1 ./ (1 + r).^t;
VPN_USD = sum(CF .* DF);
%% ----- TIR -----
TIR = irr_biseccion(CF);
%% ----- Payback -----
[payback_simple, payback_desc] = payback_metrics(CF, r);
%% ----- Precio mínimo del bulto -----
PV_bultos = sum( bultos_40kg ./ (1+r).^anio ); % años 1..n
Ing_sin_bulto_USD = Ing_ahorro_USD + Ing_lix_USD;
CF_sin_bulto = zeros(n+1,1);
CF_sin_bulto(1) = -param_econ.capex_USD;
CF_sin_bulto(2:end) = Ing_sin_bulto_USD - OPEX_total_USD;
B = sum(CF_sin_bulto .* DF);
if PV_bultos > 0
    precio_min_bulto_VPN0 = -B / PV_bultos;
else
    precio_min_bulto_VPN0 = Inf;
end
PV_OPEX = sum( OPEX_total_USD ./ (1+r).^anio );
PV_otros = sum( Ing_sin_bulto_USD ./ (1+r).^anio );
if PV_bultos > 0
    precio_min_bulto_cubre_OPEX = (PV_OPEX - PV_otros) / PV_bultos;

```

```

else
    precio_min_bulto_cubre_OPEX = Inf;
end
%% ----- Tabla anual (años 1..n) -----
tabla_1n = table( ...
    anio, masa_rec_ton, bioc_ton, bultos_40kg, ...
    lix_real_L, liq_piro_L, liq_total_L, lix_monetizable_L, ...
    Ing_bioc_USD, Ing_ahorro_USD, Ing_lix_USD, Ing_total_USD, ...
    OPEX_energia_USD, OPEX_manoobra_USD, OPEX_mantto_USD, OPEX_insumos_USD,
    OPEX_otros_USD, ...
    OPEX_total_USD, CF_anual_USD, ...
    'VariableNames', { ...
        'Anio', 'Residuos_ton', 'Biocarbon_ton', 'Bultos_40kg', ...
        'Lix_real_L', 'Liq_piro_L', 'Liq_total_L', 'Liq_monetizable_L', ...
        'Ing_bioc_USD', 'Ing_ahorro_USD', 'Ing_lix_USD', 'Ing_total_USD', ...
        'OPEX_energia_USD', 'OPEX_manoobra_USD', 'OPEX_mantto_USD', 'OPEX_insumos_USD', 'OPEX_otros_USD', ...
        'OPEX_total_USD', 'Flujo_neto_USD'});
%% ----- Tabla final 0..n -----
Anio_0n = (0:n)';
Residuos_0n = [0; masa_rec_ton];
Bultos_0n = [0; bultos_40kg];
Ing_0n = [0; Ing_total_USD];
OPEX_0n = [0; OPEX_total_USD];
Flujo_0n = CF; % ya es 0..n
Flujo_acum_0n = cumsum(Flujo_0n);
tabla_0n = table( ...
    Anio_0n, Residuos_0n, Bultos_0n, Ing_0n, OPEX_0n, Flujo_0n,
    Flujo_acum_0n, ...
    'VariableNames',
    {'Anio', 'Residuos_ton', 'Bultos_40kg', 'Ingresos_USD', 'OPEX_USD', 'Flujo_neto_USD', 'Flujo_acumulado_USD'});
%% ----- Empaquetar salida -----
res_econ = struct();
res_econ.tabla_economica_1n = tabla_1n;
res_econ.tabla_economica_0n = tabla_0n;
res_econ.resumen_financiero = table( ...
    VPN_USD, TIR, payback_simple, payback_desc, ...
    'VariableNames',
    {'VPN_USD', 'TIR', 'Payback_simple_anios', 'Payback_descontado_anios'});
res_econ.resumenPrecio_minimo = table( ...
    precio_min_bulto_cubre_OPEX, precio_min_bulto_VPN0, ...
    'VariableNames',
    {'USD_bulto_min_cubre_OPEX_PV', 'USD_bulto_min_para_VPN_cero'});
res_econ.CF = CF;
end
%% ===== AUXILIARES =====

```



```

function irr = irr_biseccion(CF)
    f = @(x) npv_from_cf(CF, x);
    a = -0.9; b = 2.0;
    fa = f(a); fb = f(b);
    if fa*fb > 0
        b = 5.0; fb = f(b);
        if fa*fb > 0
            irr = NaN; return;
        end
    end
    for k = 1:200
        m = 0.5*(a+b);
        fm = f(m);
        if abs(fm) < 1e-8
            irr = m; return;
        end
        if fa*fm < 0
            b = m; fb = fm;
        else
            a = m; fa = fm;
        end
    end
    irr = 0.5*(a+b);
end
function val = npv_from_cf(CF, r)
    t = (0:numel(CF)-1)';
    val = sum(CF ./ (1+r).^t);
end
function [pb_simple, pb_desc] = payback_metrics(CF, r)
    n = numel(CF)-1;
    cum = cumsum(CF);
    pb_simple = NaN;
    for t = 1:(n+1)
        if cum(t) >= 0
            if t == 1
                pb_simple = 0;
            else
                prev = cum(t-1);
                curr = cum(t);
                frac = (0 - prev) / (curr - prev + 1e-12);
                pb_simple = (t-2) + frac;
            end
            break;
        end
    end
    end
    td = (0:n)';
    DF = 1 ./ (1+r).^td;
    cumd = cumsum(CF .* DF);

```

```
pb_desc = NaN;
for t = 1:(n+1)
    if cumd(t) >= 0
        if t == 1
            pb_desc = 0;
        else
            prev = cumd(t-1);
            curr = cumd(t);
            frac = (0 - prev) / (curr - prev + 1e-12);
            pb_desc = (t-2) + frac;
        end
        break;
    end
end
end
```