

```

clear all;close all;clc

% Inicializar una tabla general para almacenar los resultados de todos los archivos
all_results = [];

% Inicializar listas para almacenar los promedios y desviaciones
mean_std_results = [];

% Procesar archivos desde datos1.xlsx hasta datos9.xlsx
for file_num = 1:9
    % Crear el nombre del archivo dinámicamente
    filename = ['J' num2str(file_num) '.xlsx'];
    data = readmatrix(filename);

    % Crear la figura con 5 subplots para los datos hasta deformación 0.0055
    figure;

    % Inicializar una tabla vacía para almacenar los resultados de este archivo
    table_results = cell(5,5); % 5 filas y 5 columnas

    % Nombres de las corridas (por ejemplo, J1-1, J2-1, etc.)
    for j = 1:5
        table_results{j, 1} = ['J' num2str(file_num) '-' num2str(j)];
    end

    % Inicializar listas para almacenar los valores de cada corrida (para promedio
    y desviación)
    sy_values = [];
    sut_values = [];
    e_values = [];
    elong_values = [];
    k_values = [];
    n_values = [];

    for i = 1:5
        subplot(3,2,i);

        % Obtener los datos para la corrida actual
        % deformacion = 0.5*data(:,2*i-1);
        deformacion = data(:,2*i-1);
        esfuerzo = data(:,2*i);

        % Filtrar los datos hasta deformación 0.0055
        idx = deformacion <= 0.0055;
        deformacion_filtered = deformacion(idx);
        esfuerzo_filtered = esfuerzo(idx);

        % Graficar los datos filtrados
        plot(deformacion_filtered, esfuerzo_filtered, 'k');
    end
end

```

```

hold on;

% Calcular la línea de tendencia hasta deformación 0.001

idx_trend = deformacion_filtered <= 0.001*0.9;
p = polyfit(deformacion_filtered(idx_trend), esfuerzo_filtered(idx_trend),
1);

% Graficar la línea de tendencia
yfit = polyval(p, deformacion_filtered(idx_trend));
plot(deformacion_filtered(idx_trend), yfit, '-r', 'LineWidth', 1.5);

% Crear la nueva ecuación con la misma pendiente pero que pase por (0.002,
0)

m = p(1); % La pendiente de la ecuación

% Encontrar el punto de intersección entre la línea de tendencia y la nueva
línea
[~, interseccion_idx] = min(abs(esfuerzo_filtered - (m *
(deformacion_filtered - 0.002)))));
x_interseccion = deformacion_filtered(interseccion_idx);
y_interseccion = esfuerzo_filtered(interseccion_idx);

% Definir los valores de x para la nueva línea desde 0.002 hasta el punto
de intersección
x_new = linspace(0.002, x_interseccion, 100);
y_new = m * (x_new - 0.002); % Ecuación ajustada

% Graficar la nueva línea ajustada limitada hasta el punto de intersección
plot(x_new, y_new, '--r', 'LineWidth', 1.5); % Línea verde discontinua
limitada

% Marcar el punto de intersección
plot(x_interseccion, y_interseccion, 'b.', 'MarkerSize', 20,
'MarkerFaceColor', 'b');

% Agregar el valor de sy (esfuerzo en el punto de intersección) a la tabla
table_results{i, 2} = p(1)/1000; % Almacenar la pendiente en GPa
table_results{i, 3} = y_interseccion; % Almacenar el valor de y en la
intersección (sy)

% Calcular Sut (máximo esfuerzo) y % Elongación
[~, indice_maximo] = max(esfuerzo); % Encontrar el índice del valor máximo
de esfuerzo (Sut)
sut = esfuerzo(indice_maximo); % Máximo esfuerzo
last_valid_idx = find(~isnan(deformacion), 1, 'last'); % Encontrar el
último valor válido de deformación
elongation_percent = deformacion(last_valid_idx) * 100; % Último valor
válido de deformación * 100

```

```

% Agregar Sut y % Elongación a la tabla
table_results{i, 4} = sut; % Almacenar Sut
table_results{i, 5} = elongation_percent; % Almacenar % Elongación

% Almacenar los valores para calcular promedios y desviaciones
sy_values(end+1) = y_interseccion;
sut_values(end+1) = sut;
e_values(end+1) = p(1)/1000;
elong_values(end+1) = elongation_percent;

% Etiquetas y título
xlabel('ε');
ylabel('σ');
title(['σ - ε (zona elástica) para J' num2str(file_num) '-' num2str(i)]);
hold off;
end

% Convertir los resultados de la tabla actual en formato de tabla y agregarlos
a la tabla general
table_results = cell2table(table_results, 'VariableNames', {'Probeta', 'E
[GPa]', 'Sy [MPa]', 'Sut [MPa]', '% Elongación'});
all_results = [all_results; table_results]; % Añadir los resultados a la tabla
general

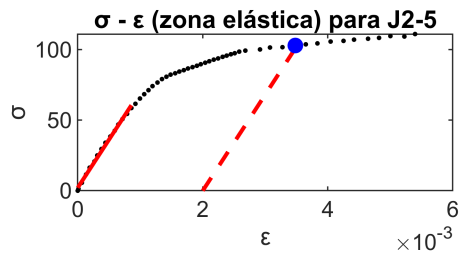
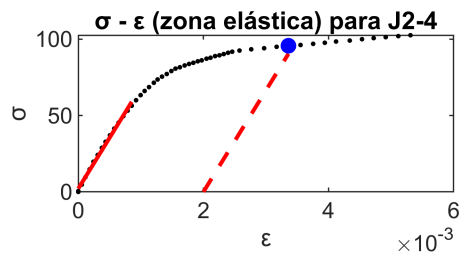
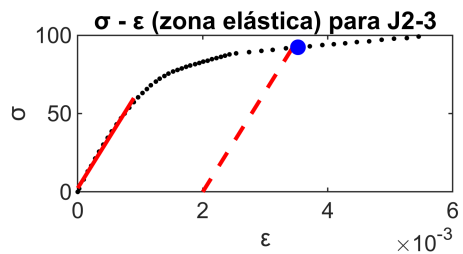
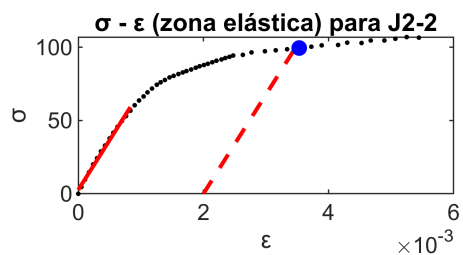
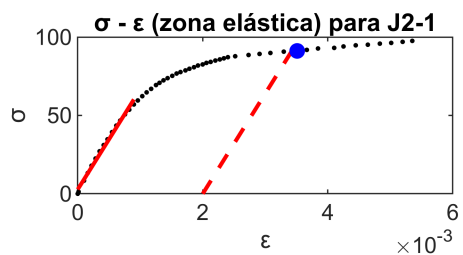
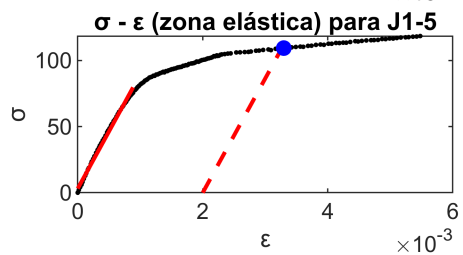
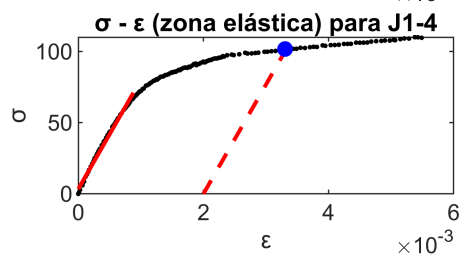
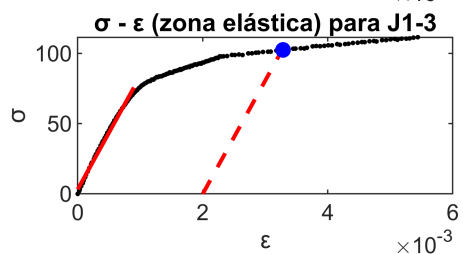
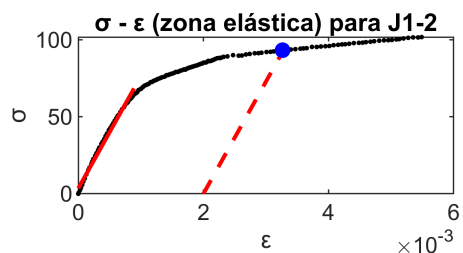
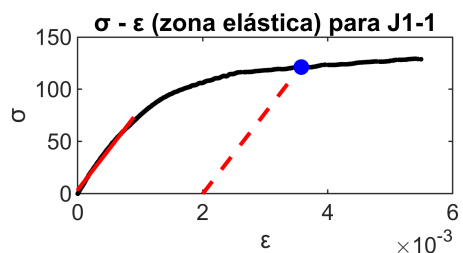
% Calcular promedios y desviaciones
mean_sy = mean(sy_values);
mean_sut = mean(sut_values);
mean_e = mean(e_values);
mean_elong = mean(elong_values);

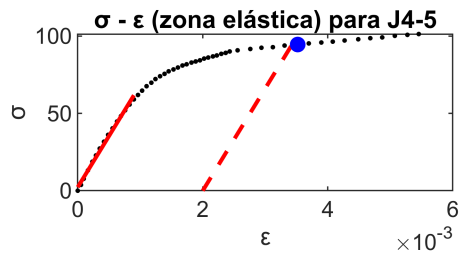
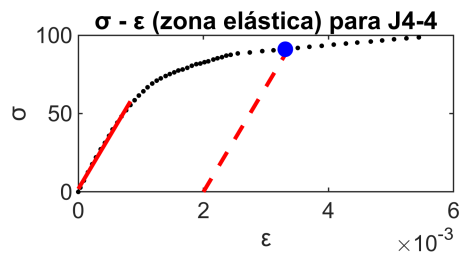
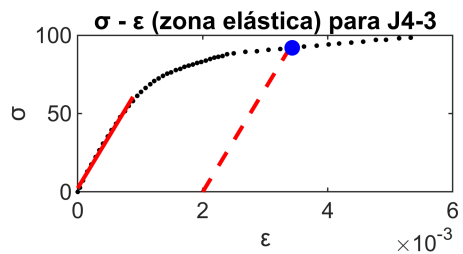
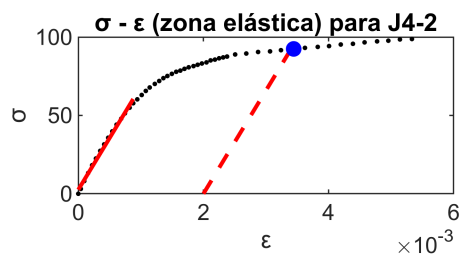
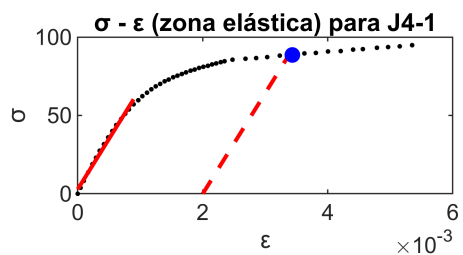
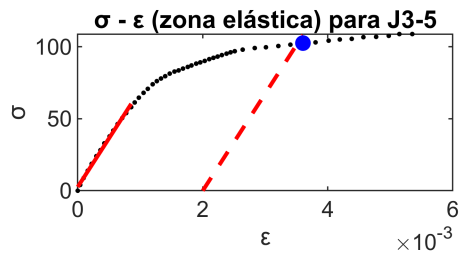
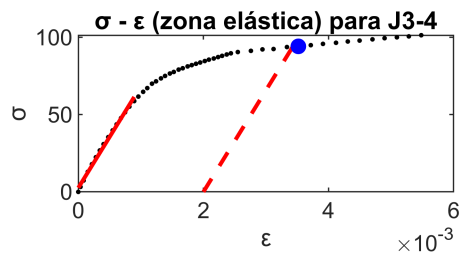
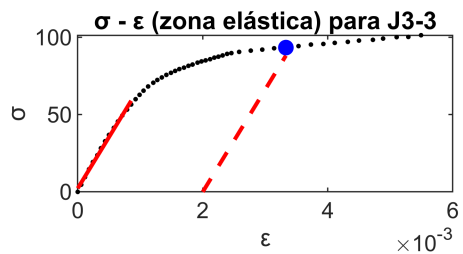
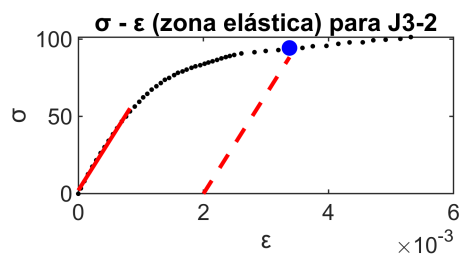
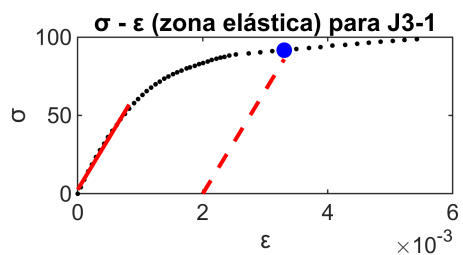
std_sy = std(sy_values);
std_sut = std(sut_values);
std_e = std(e_values);
std_elong = std(elong_values);

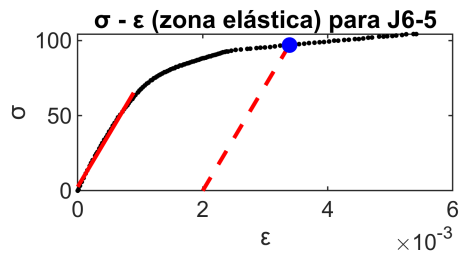
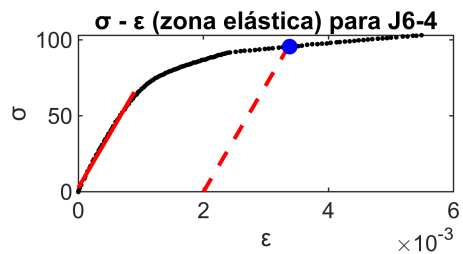
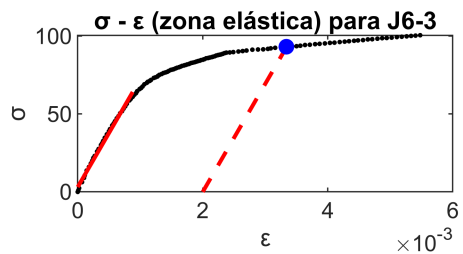
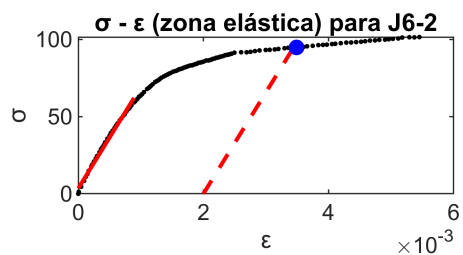
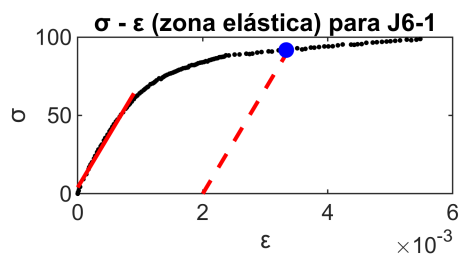
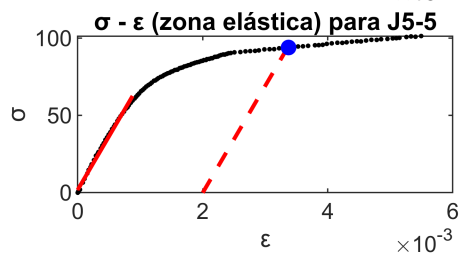
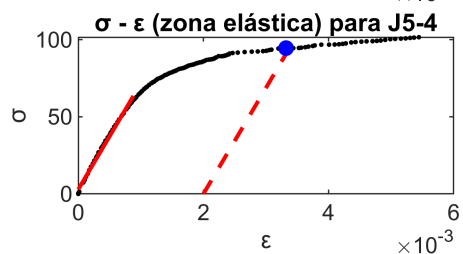
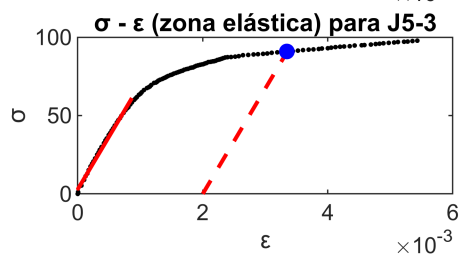
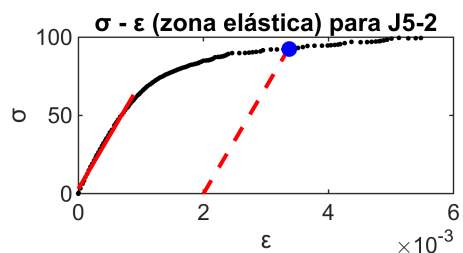
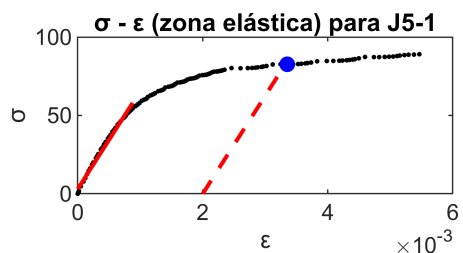
% Concatenar promedio y desviación estándar en el formato "mean ± std"
mean_std_sy = sprintf('%.2f ± %.2f', mean_sy, std_sy);
mean_std_sut = sprintf('%.2f ± %.2f', mean_sut, std_sut);
mean_std_e = sprintf('%.2f ± %.2f', mean_e, std_e);
mean_std_elong = sprintf('%.2f ± %.2f', mean_elong, std_elong);

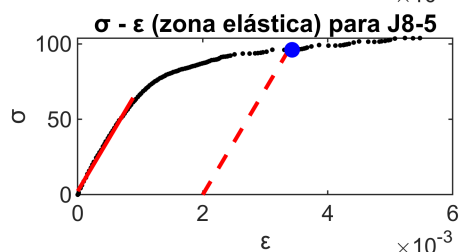
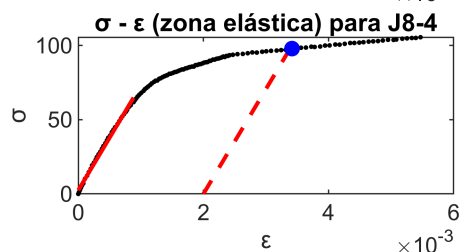
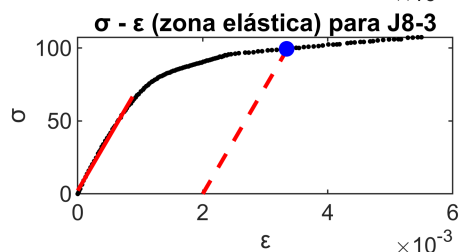
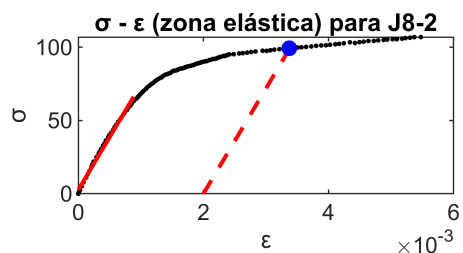
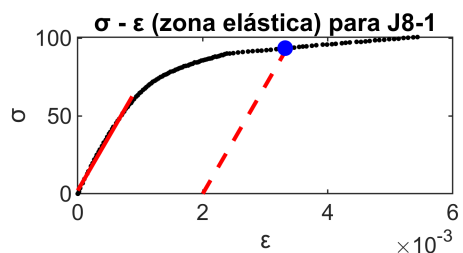
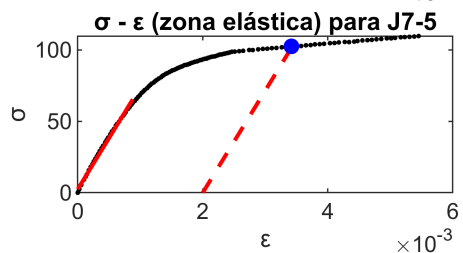
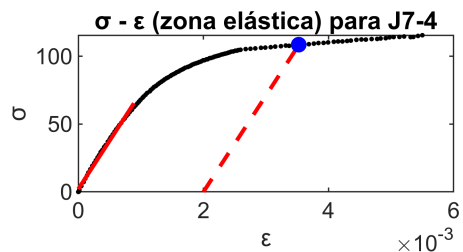
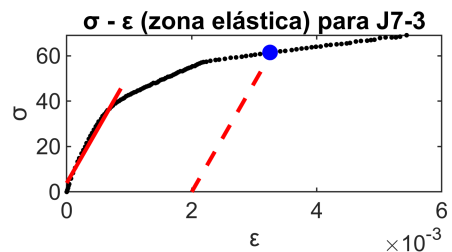
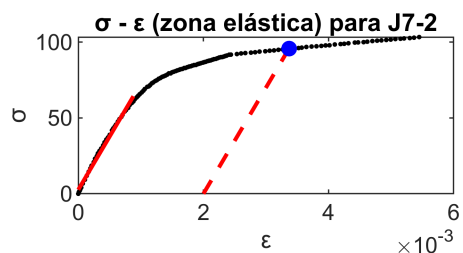
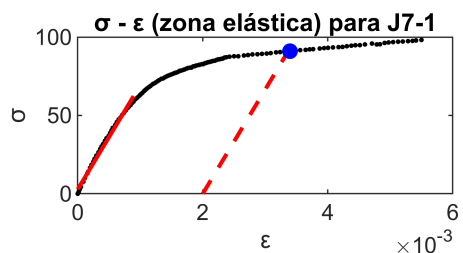
% Almacenar resultados concatenados para este Junta (J1, J2, etc.)
mean_std_results = [mean_std_results; {'J' num2str(file_num)}, mean_std_e,
mean_std_sy, mean_std_sut, mean_std_elong];%, mean_std_k, mean_std_n]];
end

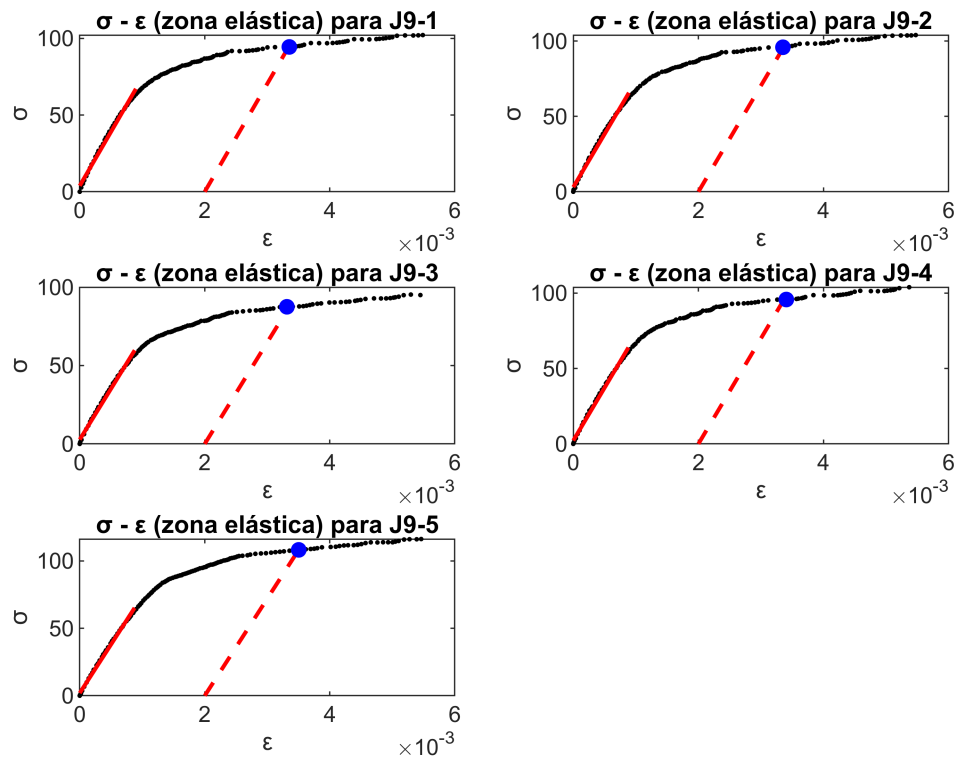
```











% Mostrar la tabla general de resultados con todas las corridas
all_results

all_results = 45x5 table

	Probeta	E [GPa]	Sy [MPa]	Sut [MPa]	% Elongación
1	'J1-1'	77.5069	121.2731	169.9502	7.7752
2	'J1-2'	72.9488	93.3239	164.0755	10.8420
3	'J1-3'	80.6644	102.4985	170.1249	9.7249
4	'J1-4'	76.8180	101.7866	169.9476	9.6119
5	'J1-5'	86.4558	109.4053	177.6034	9.3976
6	'J2-1'	63.9531	91.1584	163.4297	12.1070
7	'J2-2'	67.7676	99.4736	163.2087	11.5603
8	'J2-3'	64.0557	92.3137	161.8517	11.5587
9	'J2-4'	66.5212	95.6748	166.4358	11.7301
10	'J2-5'	68.1956	102.9935	160.0165	10.8244
11	'J3-1'	65.8940	91.6237	159.8887	11.5420
12	'J3-2'	64.2538	94.3826	160.0768	11.3408
13	'J3-3'	66.3286	93.4244	161.0635	12.3984
14	'J3-4'	65.8094	94.2139	160.5679	11.6995

	Probeta	E [GPa]	Sy [MPa]	Sut [MPa]	% Elongación
15	'J3-5'	67.4838	102.5297	161.7833	11.1406
16	'J4-1'	64.2356	88.5837	162.1621	13.2090
17	'J4-2'	65.9397	92.4409	166.6537	12.0885
18	'J4-3'	65.9434	91.9482	163.8377	12.3574
19	'J4-4'	67.0937	91.0508	163.1636	12.5384
20	'J4-5'	66.1207	94.4741	166.1748	12.0870
21	'J5-1'	62.5900	82.6421	156.0378	12.6732
22	'J5-2'	68.0157	92.3737	164.5301	12.5301
23	'J5-3'	67.2006	90.9078	160.9272	12.0928
24	'J5-4'	68.7799	94.5500	165.6423	12.6123
25	'J5-5'	69.2062	94.0145	166.9916	11.8037
26	'J6-1'	66.8105	91.7163	165.9988	12.5864
27	'J6-2'	65.7845	94.8363	169.6068	12.6729
28	'J6-3'	69.3794	93.1941	164.8400	11.8845
29	'J6-4'	70.3544	95.6108	169.1096	12.4514
30	'J6-5'	69.9944	96.9589	170.3401	12.1862
31	'J7-1'	66.6008	90.9917	160.5798	12.2481
32	'J7-2'	70.0073	95.7389	167.8693	12.1735
33	'J7-3'	48.0901	61.4767	129.3758	12.5856
34	'J7-4'	71.1066	108.6064	172.6299	12.4291
35	'J7-5'	71.5022	102.4522	172.3294	11.7986
36	'J8-1'	69.3268	93.5644	165.5886	12.3404
37	'J8-2'	71.8689	99.2654	171.6132	12.3737
38	'J8-3'	73.3536	99.4386	172.5025	11.6572
39	'J8-4'	70.6479	97.8552	168.9074	12.5532
40	'J8-5'	69.8578	96.1188	170.2868	10.8334
41	'J9-1'	70.3849	94.4291	169.9075	12.5163
42	'J9-2'	70.4847	95.9563	172.5689	12.2680
43	'J9-3'	64.8189	87.5085	160.9618	11.9548
44	'J9-4'	69.9850	95.7363	167.8551	12.6831
45	'J9-5'	71.9995	108.1709	175.3918	12.1289

```
% Convertir los resultados concatenados a una tabla
mean_std_results_table = cell2table(mean_std_results, 'VariableNames', {'Junta', 'E
[GPa]', 'Sy [MPa]', 'Sut [MPa]', '% Elongación'});
```

% Mostrar la tabla final con promedio y desviación estándar concatenados

mean_std_results_table

mean_std_results_table = 9x5 table

	Junta	E [GPa]	Sy [MPa]	Sut [MPa]	% Elongación
1	'J1'	'78.88 ± 5.05'	'105.66 ± 10.43'	'170.34 ± 4.81'	'9.47 ± 1.10'
2	'J2'	'66.10 ± 2.01'	'96.32 ± 4.94'	'162.99 ± 2.36'	'11.56 ± 0.47'
3	'J3'	'65.95 ± 1.16'	'95.23 ± 4.22'	'160.68 ± 0.77'	'11.62 ± 0.48'
4	'J4'	'65.87 ± 1.03'	'91.70 ± 2.15'	'164.40 ± 1.94'	'12.46 ± 0.46'
5	'J5'	'67.16 ± 2.67'	'90.90 ± 4.83'	'162.83 ± 4.41'	'12.34 ± 0.38'
6	'J6'	'68.46 ± 2.04'	'94.46 ± 2.05'	'167.98 ± 2.41'	'12.36 ± 0.32'
7	'J7'	'65.46 ± 9.90'	'91.85 ± 18.25'	'160.56 ± 18.10'	'12.25 ± 0.30'
8	'J8'	'71.01 ± 1.62'	'97.25 ± 2.45'	'169.78 ± 2.71'	'11.95 ± 0.71'
9	'J9'	'69.53 ± 2.75'	'96.36 ± 7.45'	'169.34 ± 5.47'	'12.31 ± 0.29'

% Extraer los nombres de los juntas y propiedades de la tabla de resultados

% juntas = mean_std_results_table.Junta;

E_vals = mean_std_results_table('E [GPa]');

Sy_vals = mean_std_results_table('Sy [MPa]');

Sut_vals = mean_std_results_table('Sut [MPa]');

elong_vals = mean_std_results_table('% Elongación');

% Separar los valores medios y desviaciones estándar de cada propiedad

E_means = [];

Sy_means = [];

Sut_means = [];

elong_means = [];

E_stds = [];

Sy_stds = [];

Sut_stds = [];

elong_stds = [];

% Descomponer los valores concatenados en medios y desviaciones estándar

for i = 1:length(E_vals)

 E_parts = sscanf(E_vals{i}, '%f ± %f');

 Sy_parts = sscanf(Sy_vals{i}, '%f ± %f');

 Sut_parts = sscanf(Sut_vals{i}, '%f ± %f');

 elong_parts = sscanf(elong_vals{i}, '%f ± %f');

 E_means(i) = E_parts(1);

 E_stds(i) = E_parts(2);

 Sy_means(i) = Sy_parts(1);

 Sy_stds(i) = Sy_parts(2);

```

    Sut_means(i) = Sut_parts(1);
    Sut_stds(i) = Sut_parts(2);

    elong_means(i) = elong_parts(1);
    elong_stds(i) = elong_parts(2);
end

% Definir las velocidades de rotación y avance correspondientes a cada junta
vel_rotacion = [620, 620, 620, 720, 720, 720, 520, 520, 520]; % en rpm
vel_avance = [30, 35, 25, 30, 35, 25, 30, 35, 25]; % en mm/min

% Crear la malla de velocidades de rotación y avance para interpolación
[vel_rot_grid, vel_avance_grid] = meshgrid(unique(vel_rotacion),
unique(vel_avance));

% Interpolación para obtener valores continuos
E_interp = griddata(vel_rotacion, vel_avance, E_means, vel_rot_grid,
vel_avance_grid, 'cubic');
Sy_interp = griddata(vel_rotacion, vel_avance, Sy_means, vel_rot_grid,
vel_avance_grid, 'cubic');
Sut_interp = griddata(vel_rotacion, vel_avance, Sut_means, vel_rot_grid,
vel_avance_grid, 'cubic');
elong_interp = griddata(vel_rotacion, vel_avance, elong_means, vel_rot_grid,
vel_avance_grid, 'cubic');

% Crear una figura para los mapas de calor
figure;

% 1. Mapa de calor para el Módulo de Elasticidad (E)
subplot(2,2,1);
contourf(vel_rot_grid, vel_avance_grid, E_interp, 20, 'LineColor', 'none'); % Mapa
de calor continuo
colormap('jet'); % Cambiar el colormap a 'jet'
colorbar;
xlabel('V_{rot} (rpm)');
ylabel('V_{av} (mm/min)');
title('E');

% 2. Mapa de calor para el Esfuerzo de Fluencia (Sy)
subplot(2,2,2);
contourf(vel_rot_grid, vel_avance_grid, Sy_interp, 20, 'LineColor', 'none'); %
Mapa de calor continuo
colormap('jet');
colorbar;
xlabel('V_{rot} (rpm)');
ylabel('V_{av} (mm/min)');
title('S_y');

% 3. Mapa de calor para el Esfuerzo Último (Sut)

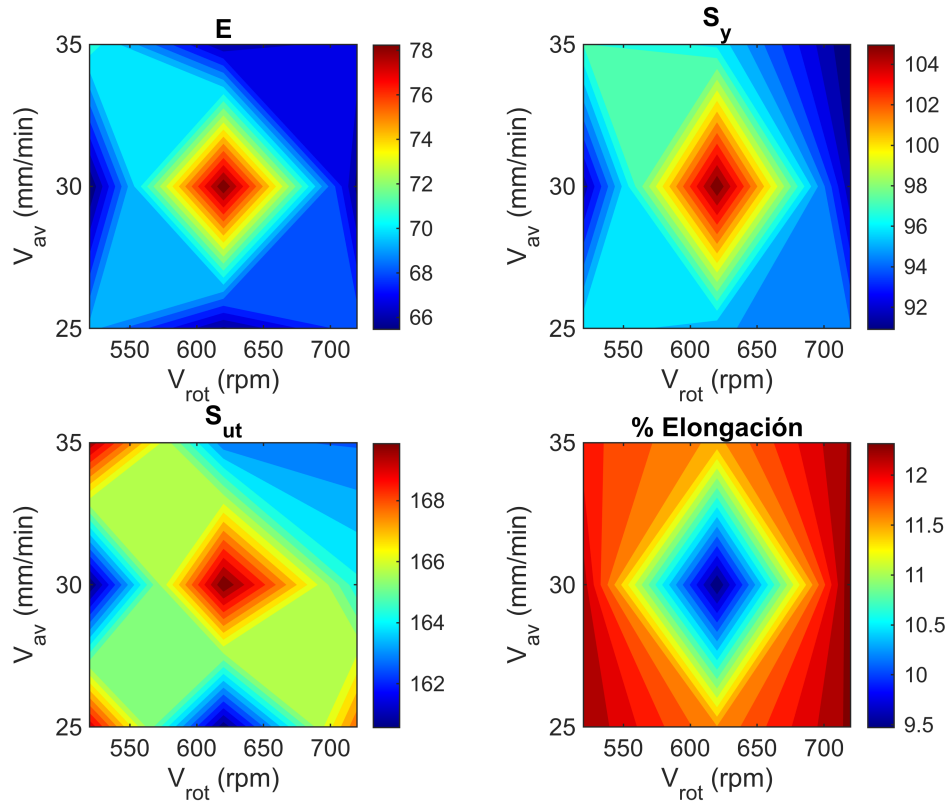
```

```

subplot(2,2,3);
contourf(vel_rot_grid, vel_avance_grid, Sut_interp, 20, 'LineColor', 'none'); %
Mapa de calor continuo
colormap('jet');
colorbar;
xlabel('V_{rot} (rpm)');
ylabel('V_{av} (mm/min)');
title('S_{ut}');

% 4. Mapa de calor para el % de Elongación
subplot(2,2,4);
contourf(vel_rot_grid, vel_avance_grid, elong_interp, 20, 'LineColor', 'none'); %
Mapa de calor continuo
colormap('jet');
colorbar;
xlabel('V_{rot} (rpm)');
ylabel('V_{av} (mm/min)');
title('% Elongación');

```



```

% Datos de las juntas y propiedades
juntas = mean_std_results_table.Junta;
E_vals = mean_std_results_table('E [GPa]');
Sy_vals = mean_std_results_table('Sy [MPa]');
Sut_vals = mean_std_results_table('Sut [MPa]');
elong_vals = mean_std_results_table('% Elongación');

```

```

% Separar los valores medios y desviaciones estándar de cada propiedad
E_means = [];
Sy_means = [];
Sut_means = [];
elong_means = [];
E_stds = [];
Sy_stds = [];
Sut_stds = [];
elong_stds = [];

% Descomponer los valores concatenados en medios y desviaciones estándar
for i = 1:length(E_vals)
    E_parts = sscanf(E_vals{i}, '%f ± %f');
    Sy_parts = sscanf(Sy_vals{i}, '%f ± %f');
    Sut_parts = sscanf(Sut_vals{i}, '%f ± %f');
    elong_parts = sscanf(elong_vals{i}, '%f ± %f');

    E_means(i) = E_parts(1);
    E_stds(i) = E_parts(2);

    Sy_means(i) = Sy_parts(1);
    Sy_stds(i) = Sy_parts(2);

    Sut_means(i) = Sut_parts(1);
    Sut_stds(i) = Sut_parts(2);

    elong_means(i) = elong_parts(1);
    elong_stds(i) = elong_parts(2);
end

% Configurar el espacio adicional en el eje X
x_spacing = 0.5; % Ajuste de espacio en los bordes del eje X
x_values = 1:3; % Número de columnas de subplots por fila (juntas)

% Crear una figura para los subplots (4 filas, 3 columnas)
figure;

% Colores para cada propiedad (colores contrastantes)
colors = {
    [0.2, 0.3, 1], % Azul contrastante para E [GPa]
    [0.7, 0.2, 1], % Morado fuerte para Sy [MPa]
    [0, 0.5, 0.7], % Azul verdoso para Sut [MPa]
    [0.8, 0, 0.5] % Morado oscuro para % Elongación
};

% 1. Subplot fila 1: Módulo de Elasticidad (E)
for i = 1:3
    subplot(4,3,i); % Crear el subplot en la posición correspondiente
    errorbar(x_values, E_means((i-1)*3+1:i*3), E_stds((i-1)*3+1:i*3), '.', 'Color',
    colors{1}, 'MarkerSize', 12, 'LineWidth', 1);

```

```

hold on;
% Generar una línea spline punteada para unir los puntos medios
xx = linspace(1, 3, 100);
yy = spline(x_values, E_means((i-1)*3+1:i*3), xx);
plot(xx, yy, '--', 'Color', colors{1}, 'LineWidth', 1); % Línea punteada
hold off;
set(gca, 'XTick', x_values, 'XTickLabel', juntas((i-1)*3+1:i*3));
xlim([1 - x_spacing, length(x_values) + x_spacing]);
ylim([min(E_means((i-1)*3+1:i*3) - E_stds((i-1)*3+1:i*3)) * 0.95,
max(E_means((i-1)*3+1:i*3) + E_stds((i-1)*3+1:i*3)) * 1.05]);
xlabel('Junta');
ylabel('E [GPa]');
title('Módulo de Elasticidad (E)');
end

% 2. Subplot fila 2: Esfuerzo de Fluencia (Sy)
for i = 1:3
    subplot(4,3,i+3); % Crear el subplot en la fila 2
    errorbar(x_values, Sy_means((i-1)*3+1:i*3), Sy_stds((i-1)*3+1:i*3),
    '.', 'Color', colors{2}, 'MarkerSize', 12, 'LineWidth', 1);
    hold on;
    % Generar una línea spline punteada para unir los puntos medios
    yy = spline(x_values, Sy_means((i-1)*3+1:i*3), xx);
    plot(xx, yy, '--', 'Color', colors{2}, 'LineWidth', 1); % Línea punteada
    hold off;
    set(gca, 'XTick', x_values, 'XTickLabel', juntas((i-1)*3+1:i*3));
    xlim([1 - x_spacing, length(x_values) + x_spacing]);
    ylim([min(Sy_means((i-1)*3+1:i*3) - Sy_stds((i-1)*3+1:i*3)) * 0.95,
max(Sy_means((i-1)*3+1:i*3) + Sy_stds((i-1)*3+1:i*3)) * 1.05]);
    xlabel('Junta');
    ylabel('Sy [MPa]');
    title('Esfuerzo de Fluencia (Sy)');
end

% 3. Subplot fila 3: Esfuerzo Último (Sut)
for i = 1:3
    subplot(4,3,i+6); % Crear el subplot en la fila 3
    errorbar(x_values, Sut_means((i-1)*3+1:i*3), Sut_stds((i-1)*3+1:i*3),
    '.', 'Color', colors{3}, 'MarkerSize', 12, 'LineWidth', 1);
    hold on;
    % Generar una línea spline punteada para unir los puntos medios
    yy = spline(x_values, Sut_means((i-1)*3+1:i*3), xx);
    plot(xx, yy, '--', 'Color', colors{3}, 'LineWidth', 1); % Línea punteada
    hold off;
    set(gca, 'XTick', x_values, 'XTickLabel', juntas((i-1)*3+1:i*3));
    xlim([1 - x_spacing, length(x_values) + x_spacing]);
    ylim([min(Sut_means((i-1)*3+1:i*3) - Sut_stds((i-1)*3+1:i*3)) * 0.95,
max(Sut_means((i-1)*3+1:i*3) + Sut_stds((i-1)*3+1:i*3)) * 1.05]);
    xlabel('Junta');
    ylabel('Sut [MPa]');
end

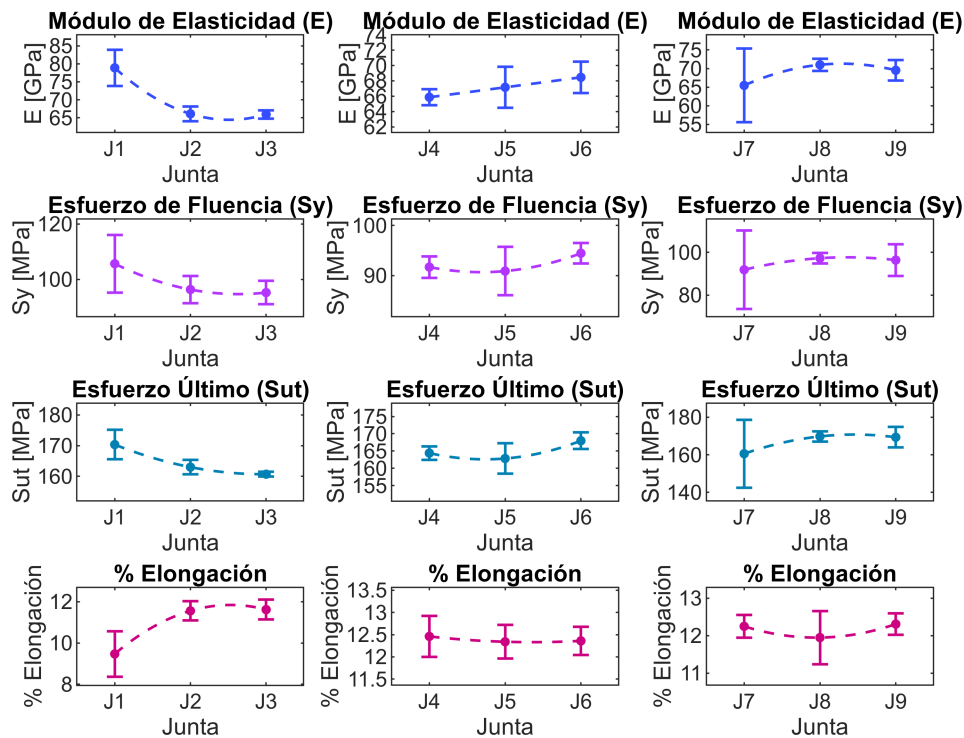
```

```

title('Esfuerzo Último (Sut)');
end

% 4. Subplot fila 4: % Elongación
for i = 1:3
    subplot(4,3,i+9); % Crear el subplot en la fila 4
    errorbar(x_values, elong_means((i-1)*3+1:i*3), elong_stds((i-1)*3+1:i*3),
    '.', 'Color', colors{4}, 'MarkerSize', 12, 'LineWidth', 1);
    hold on;
    % Generar una línea spline punteada para unir los puntos medios
    yy = spline(x_values, elong_means((i-1)*3+1:i*3), xx);
    plot(xx, yy, '--', 'Color', colors{4}, 'LineWidth', 1); % Línea punteada
    hold off;
    set(gca, 'XTick', x_values, 'XTickLabel', juntas((i-1)*3+1:i*3));
    xlim([1 - x_spacing, length(x_values) + x_spacing]);
    ylim([min(elong_means((i-1)*3+1:i*3) - elong_stds((i-1)*3+1:i*3)) * 0.95,
    max(elong_means((i-1)*3+1:i*3) + elong_stds((i-1)*3+1:i*3)) * 1.05]);
    xlabel('Junta');
    ylabel('% Elongación');
    title('% Elongación');
end

```



```
clear all; close all; clc;
```

```
% Inicializar una tabla general para almacenar los resultados de todos los archivos
all_results = [];
```

```

mean_std_results = [];
sy_values_all = [];
sut_values_all = [];
e_values_all = [];
elong_values_all = [];

% Variables para el análisis factorial (3 niveles para A y B)
A = [30 30 30 30 30 35 35 35 35 35 25 25 25 25 25 ...
      30 30 30 30 30 35 35 35 35 35 25 25 25 25 25 ...
      30 30 30 30 30 35 35 35 35 35 25 25 25 25 25];
B = repelem([620, 720, 520], 5 * 3); % Nivel de Velocidad de rotación (3 niveles)

for file_num = 1:9
    % Crear el nombre del archivo dinámicamente
    filename = ['J' num2str(file_num) '.xlsx'];
    data = readmatrix(filename);

    sy_values = [];
    sut_values = [];
    e_values = [];
    elong_values = [];

    for i = 1:5
        deformacion = data(:, 2 * i - 1);
        esfuerzo = data(:, 2 * i);

        % Filtrar los datos hasta deformación 0.0055
        idx = deformacion <= 0.0055;
        deformacion_filtered = deformacion(idx);
        esfuerzo_filtered = esfuerzo(idx);

        % Calcular la línea de tendencia hasta deformación 0.001
        idx_trend = deformacion_filtered <= 0.001 * 0.9;
        p = polyfit(deformacion_filtered(idx_trend), esfuerzo_filtered(idx_trend),
1);

        % Encontrar el punto de intersección
        m = p(1);
        [~, interseccion_idx] = min(abs(esfuerzo_filtered - (m *
(deformacion_filtered - 0.002))));
        y_interseccion = esfuerzo_filtered(interseccion_idx);

        % Calcular Sut (máximo esfuerzo) y % Elongación
        [~, indice_maximo] = max(esfuerzo);
        sut = esfuerzo(indice_maximo);
        last_valid_idx = find(~isnan(deformacion), 1, 'last');
        elongation_percent = deformacion(last_valid_idx) * 100;

        % Almacenar los valores
        sy_values(end + 1) = y_interseccion;

```



```

        sut_values(end + 1) = sut;
        e_values(end + 1) = p(1) / 1000;
        elong_values(end + 1) = elongation_percent;
    end

    % Almacenar los resultados de las juntas
    sy_values_all = [sy_values_all, sy_values];
    sut_values_all = [sut_values_all, sut_values];
    e_values_all = [e_values_all, e_values];
    elong_values_all = [elong_values_all, elong_values];
end

% Convertir los resultados concatenados a una tabla para E, Sy, Sut, %Elong
variables = {'E', 'Sy', 'Sut', '% Elongación'};
responses = {e_values_all', sy_values_all', sut_values_all', elong_values_all'};

for var_idx = 1:length(variables)
    Y = responses{var_idx}; % Variable de respuesta actual

    % Modelo lineal general (GLM) para analizar los efectos principales y la
    interacción
    mdl = fitlm([A', B'], Y, 'linear', 'CategoricalVars', [1, 2]);

    % Mostrar los coeficientes y la tabla ANOVA

    fprintf('.....\n')
    fprintf('\n')
    disp(['Resultados para ', variables{var_idx}]);
    fprintf('\n')
    disp(mdl.Coefficients);
    disp(anova(mdl, 'summary'));

    % Crear una figura con subplots para el efecto de A, B y A*B
    figure;

    % 1. Gráfico del efecto principal de A
    subplot(2,2,1); % Primer subplot
    main_effect_A = grpstats(Y, A, 'mean');
    x_vals_A = [25 30 35];
    x_dense_A = linspace(min(x_vals_A), max(x_vals_A), 100);
    y_spline_A = spline(x_vals_A, main_effect_A, x_dense_A);
    plot(x_dense_A, y_spline_A, '-', 'LineWidth', 1); hold on;
    plot(x_vals_A, main_effect_A, '.k', 'MarkerSize', 20);
    xlim([23 37]);
    ylim([0.99*min(main_effect_A) 1.01*max(main_effect_A)]);
    set(gca, 'XTick', [25 30 35]);
    xlabel('Velocidad de Avance');
    ylabel(variables{var_idx});
    title('Efecto Principal de A');

```

```

% 2. Gráfico del efecto principal de B
subplot(2,2,2); % Segundo subplot
main_effect_B = grpstats(Y, B, 'mean');
x_vals_B = [520 620 720];
x_dense_B = linspace(min(x_vals_B), max(x_vals_B), 100);
y_spline_B = spline(x_vals_B, main_effect_B, x_dense_B);
plot(x_dense_B, y_spline_B, '-', 'LineWidth', 1); hold on;
plot(x_vals_B, main_effect_B, '.k', 'MarkerSize', 20);
xlim([500 740]);
ylim([0.99*min(main_effect_B) 1.01*max(main_effect_B)]);
set(gca, 'XTick', [520 620 720]);
xlabel('Velocidad de Rotación');
ylabel(variables{var_idx});
title('Efecto Principal de B');

% 3. Gráfico de la interacción A*B (Ocupando [3 4])
subplot(2,2,[3 4]); % Ocupa el espacio de [3 4]

% Generar combinaciones únicas de A y B
unique_groups = unique([A', B'], 'rows');
interaction_effects = zeros(size(unique_groups, 1), 1);

for i = 1:size(unique_groups, 1)
    interaction_effects(i) = mean(Y(A == unique_groups(i,1) & B ==
unique_groups(i,2)));
end

% Ajustar el gráfico para representar una superficie
[A_levels, B_levels] = meshgrid([25 30 35], [520 620 720]);
surf(A_levels, B_levels, reshape(interaction_effects, [3,3]));
xlabel('Velocidad de Avance');
ylabel('Velocidad de Rotación');
zlabel(variables{var_idx});
title('Interacción A * B');
set(gca, 'YTick', [520 620 720]);
grid on;
shading interp; % Suaviza el gráfico
colorbar;
hold off;

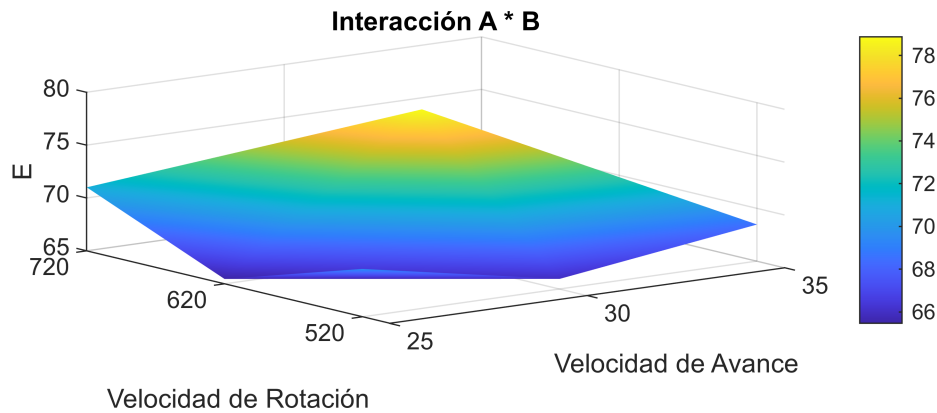
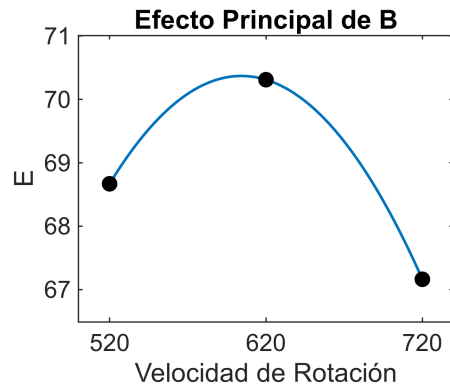
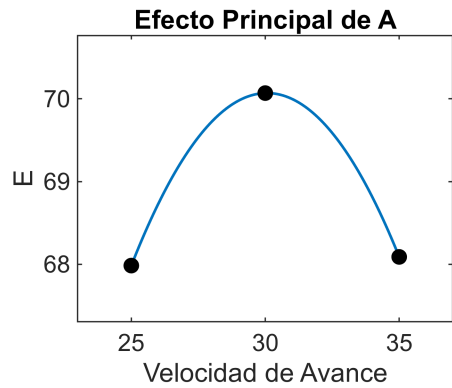
end

```

.....
Resultados para E

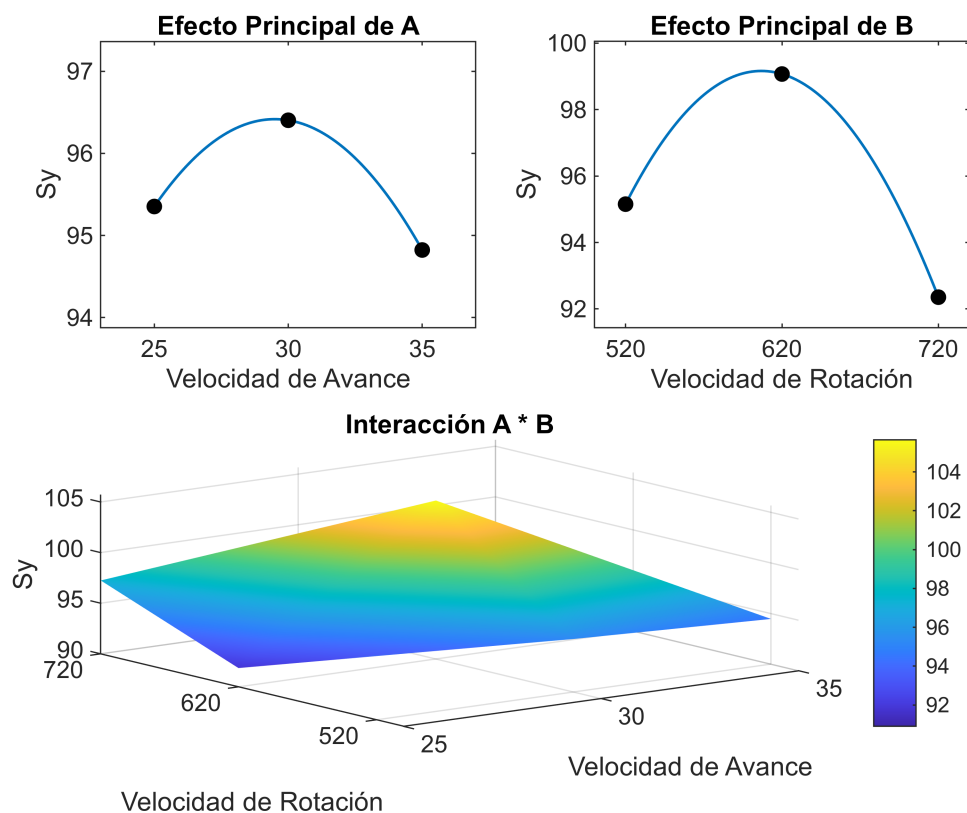
	Estimate	SE	tStat	pValue
(Intercept)	67.939	1.8366	36.992	1.4817e-32
x1_30	2.0846	2.0119	1.0361	0.30637
x1_35	0.10499	2.0119	0.052187	0.95864

x2_620	1.6414	2.0119	0.81588	0.4194	
x2_720	-1.5058	2.0119	-0.74843	0.45858	
	SumSq	DF	MeanSq	F	pValue
<hr/>					
Total	1330	44	30.227		
Model	115.71	4	28.927	0.95287	0.44379
Residual	1214.3	40	30.358		
. Lack of fit	608.79	4	152.2	9.0487	3.6426e-05
. Pure error	605.51	36	16.82		



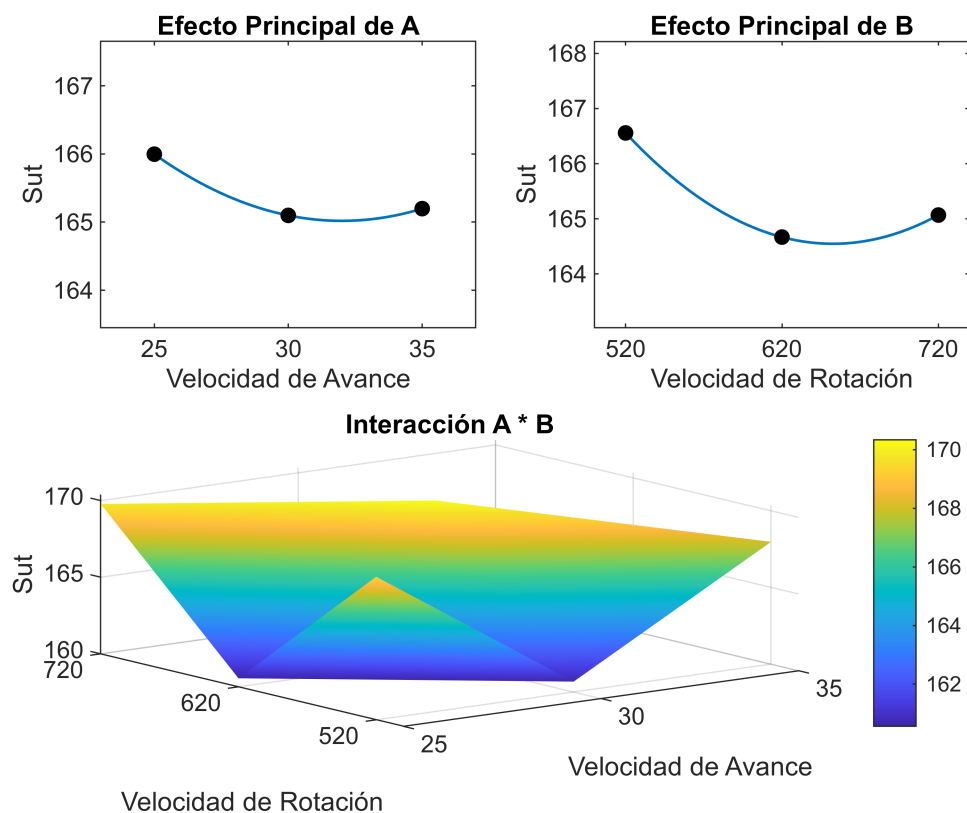
.....
Resultados para Sy

	Estimate	SE	tStat	pValue	
<hr/>					
(Intercept)	94.98	2.7577	34.441	2.371e-31	
x1_30	1.0506	3.0209	0.34778	0.72983	
x1_35	-0.52981	3.0209	-0.17538	0.86167	
x2_620	3.9177	3.0209	1.2969	0.20211	
x2_720	-2.8005	3.0209	-0.92702	0.35947	
	SumSq	DF	MeanSq	F	pValue
<hr/>					
Total	3098.9	44	70.429		
Model	361.04	4	90.26	1.3187	0.27959
Residual	2737.8	40	68.446		
. Lack of fit	427.52	4	106.88	1.6654	0.17938
. Pure error	2310.3	36	64.175		



.....
Resultados para Sut

	Estimate	SE	tStat	pValue	
(Intercept)	167.12	2.5167	66.407	1.4941e-42	
x1_30	-0.89884	2.7569	-0.32604	0.7461	
x1_35	-0.79938	2.7569	-0.28996	0.77335	
x2_620	-1.8896	2.7569	-0.6854	0.49705	
x2_720	-1.4901	2.7569	-0.54051	0.59184	
	SumSq	DF	MeanSq	F	pValue
Total	2317.2	44	52.663		
Model	37.036	4	9.2591	0.16243	0.95612
Residual	2280.1	40	57.003		
. Lack of fit	587.7	4	146.93	3.1253	0.026371
. Pure error	1692.4	36	47.011		



.....

Resultados para % Elongación

	Estimate	SE	tStat	pValue
(Intercept)	12.454	0.24966	49.883	1.2193e-37
x1_30	-0.70581	0.27349	-2.5807	0.013634
x1_35	-0.14688	0.27349	-0.53706	0.5942
x2_620	-1.286	0.27349	-4.7023	3.0416e-05
x2_720	0.21531	0.27349	0.78726	0.43577
	SumSq	DF	MeanSq	F
				pValue
Total	46.371	44	1.0539	
Model	23.932	4	5.9831	10.665
Residual	22.439	40	0.56098	
. Lack of fit	11.235	4	2.8088	9.0251
. Pure error	11.204	36	0.31122	

