

Punto 3 parcial campo electromagnéticos
Andrés Bernal - 7003748
Jerónimo Murcia 7003850

Código Genérico

```
% Parámetros de la malla
nx = 100; % Número de puntos en x
ny = 100; % Número de puntos en y
dx = 1; % Tamaño de la celda en x
dy = 1; % Tamaño de la celda en y

% Condiciones de contorno
u_top = 100; % Valor en la parte superior
u_bottom = 0; % Valor en la parte inferior
u_left = 0; % Valor en el lado izquierdo
u_right = 0; % Valor en el lado derecho

% Inicializar la matriz de la solución
u = zeros(nx, ny);

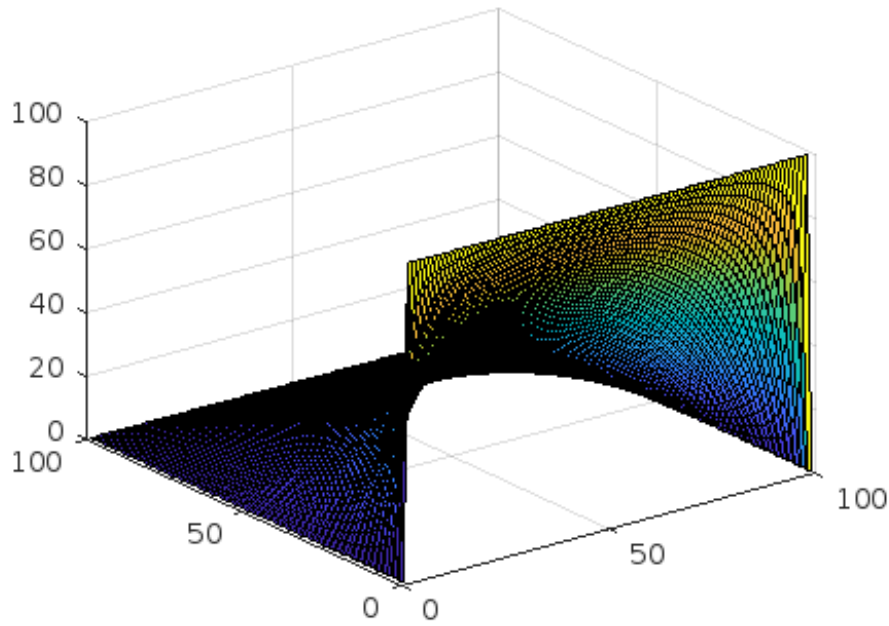
% Asignar condiciones de contorno
u(1,:) = u_top;
u(end,:) = u_bottom;
u(:,1) = u_left;
u(:,end) = u_right;

% Iterar hasta que la solución converja
error = 1; % Error inicial
tol = 1e-6; % Tolerancia
while error > tol
    % Copiar la matriz anterior
    u_old = u;

    % Actualizar la solución para cada punto
    for i = 2:nx-1
        for j = 2:ny-1
            u(i,j) = (u_old(i+1,j) + u_old(i-1,j) + u_old(i,j+1) + u_old(i,j-1)) / 4;
        end
    end

    % Calcular el error
    error = max(max(abs(u - u_old)));
end

% Visualizar la solución
surf(u);
```



Código Solución D7.6

```
% Parámetros de la malla
nx = 50; % Número de puntos en x
ny = 50; % Número de puntos en y
dx = 1; % Tamaño de la celda en x
dy = 1; % Tamaño de la celda en y

% Condiciones de contorno
u_top = 100; % Valor en la parte superior
u_bottom = 40; % Valor en la parte inferior
u_left = 0; % Valor en el lado izquierdo
u_right = 100; % Valor en el lado derecho

% Inicializar la matriz de la solución
u = zeros(nx, ny);

% Agregar potencial irregular
for i = 1:nx
    for j = 1:ny
        if i < nx/4 || i > 3*nx/4 || j < ny/4 || j > 3*ny/4
            u(i,j) = 0;
        else
            u(i,j) = 100;
        end
    end
end

% Asignar condiciones de contorno
```

```

u(1,:) = u_top;
u(end,:) = u_bottom;
u(:,1) = u_left;
u(:,end) = u_right;

% Iterar hasta que la solución converja
error = 1; % Error inicial
tol = 1e-6; % Tolerancia
while error > tol
    % Copiar la matriz anterior
    u_old = u;

    % Actualizar la solución para cada punto
    for i = 2:nx-1
        for j = 2:ny-1
            u(i,j) = (u_old(i+1,j) + u_old(i-1,j) + u_old(i,j+1) + u_old(i,j-1)) / 4;
        end
    end

    % Calcular el error
    error = max(max(abs(u - u_old)));
end

% Visualizar la solución
surf(u);

```

