Apéndice 2: Algoritmo Matching Pursuit (MP)

```
%-----------------------------------------------------------------------%
%IMPLEMENTACIÓN DEL ALGORITMO MATCHING PURSUIT PARA LA RECONSTUCCIÓN
%DE SEÑALES SPARSE
%-----------------------------------------------------------------------%

function MP

n = 1024;                       % Longitud de la señal x
m = 100;                        % Número de proyecciones aleatorias
k = 20;                         % Nivel de escasez de la señal
x = zeros(n,1);                 % Señal x = 0
q = 1+randint(1,k,n);           % Posiciones de las componentes
                               % distintas de cero
x(q(1:k)) = 2*randn(k,1)-1;     % Señal sparse generada

% Generación del diccionario

psi = eye(n,n);                 % Diccionario

% Proyecciones aleatorias de la señal x

phi = 2*randint(m,n)-1;         % Matriz de proyecciones aleatorias
y = phi*x;                      % Proyecciones aleatorias de la señal

% Algoritmo Matching Pursuit

r = zeros(m,2);                  % Residuo en las iteraciones t y t-1
r(:,1) = y;                      % Residuo igual a las proyecciones
s = zeros(length(psi(1,:)),1);   % Estimado de la señal igual a cero
V = phi*psi;                     % Diccionario holográfico
epsilon = 1e-16;
t = 0;                           % Contador de iteraciones igual a 0
error = 0;
Vn = sqrt(sum(V.^2));            % Norma de las columnas de V

while(t < n  &&  norm(r(:,1))^2 > epsilon*norm(y)^2)  % Decisión

    nt = abs (V'*r(:,1)./Vn');  % Proyecciones del residuo sobre los
                                % átomos del diccionario holográfico

    [val pos] = max(nt);        % Valor y posición de la máxima
                                % proyección
    rnt = dot(r(:,1),V(:,pos))/Vn(pos)^2;   % Contribución del átomo

    r(:,2) = r(:,1)-rnt*V(:,pos);% Actualización del residuo

    s(pos,1) = s(pos,1)+rnt;     % Actualización del coeficiente

    t = t+1;                     % Incremento del contador de
                                 % iteraciones
    r(:,1) = r(:,2);             % Residuo anterior igual al actual

end

xest = psi*s;                    % Estimación de la señal original
```