

```
% Ventanea la señal utilizando ventanas solapadas (cada media ventana)
% de la longitud y el tipo especificado. Para no dejar muestras afuera,
% ventanea sobre una copia de la señal con ceros agregados. Sirve
% tanto para señales como vectores fila y columna, pero devuelve
% una matriz cuyas filas contienen las ventanas.

function [ventanas]=ventanearsenal(senal, muestrasVentana, tipoVentana)
    ventanas=[];    % guarda las ventanas
    % calcula el numero de ventanas solapadas que pueden tomarse de la señal
    nVentanas=floor(length(senal)/floor(muestrasVentana/2))-1;

    % calculamos la cantidad de ceros que van al final
    nCeros=(nVentanas*floor(muestrasVentana/2)+muestrasVentana)-length(senal);

    % rellena con ceros al final de la señal para no dejar muestras afuera
    % rellenamos segun la señal sea vector fila o vector columna
    [nfilas, ncolumnas]=size(senal);
    if(nfilas==1)    % señal es un vector fila
        senal=[senal zeros(1,nCeros)];
    else            % señal es un vector columna
        senal=[senal; zeros(nCeros,1)];
    end

    % realiza el ventaneo
    for i=0:nVentanas
        temp=senal(i*floor(muestrasVentana/2)+1:i*floor(muestrasVentana/2)+muestrasVentana);
        % aplica ventana suavizada
        if(nfilas==1)    % señal es un vector fila
            temp=temp.*feval(tipoVentana, length(temp))';
            ventanas=[ventanas; temp];
        else            % señal es un vector columna
            temp=temp.*feval(tipoVentana, length(temp));
            ventanas=[ventanas; temp'];
        end
    end
end
```