

GUÍA DE USO PARA LA PUBLICACIÓN DE LABORATORIOS VIRTUALES EN MATLAB APP DESIGNER.

- El Proyecto “Laboratorios Virtuales”.

Estos laboratorios están basados en la herramienta MATLAB App Designer de MathWorks. Esta herramienta genera ficheros .mlapp que una vez compiladas y publicadas con MathWorks MATLAB Web App Server nos permite visualizar en un navegador web documentos interactivos creados con MATLAB.

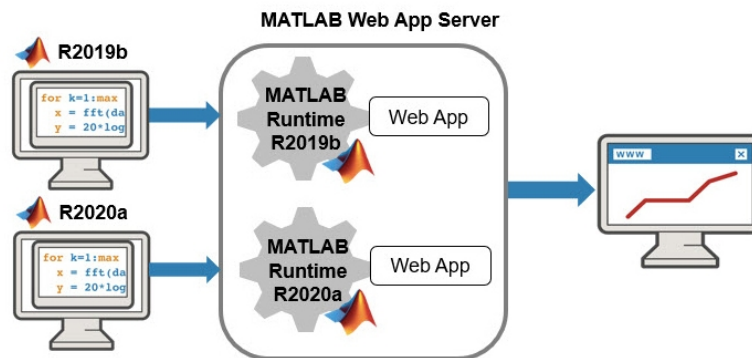


Figura 1.

Para poder visualizar las simulaciones, éstas se deben crear con el programa Matlab R2019b o superior (archivo *.mlapp), la UPV dispone de licencia de campus para su uso entre el profesorado.

- Proceso de envío del laboratorio.

Para enviar un laboratorio se **deberá comprimir en un fichero ZIP** el fichero .mlapp junto con los ficheros auxiliares, si fueran necesarios. Es importante también rellenar todos los apartados solicitados en el formulario recibido para la correcta publicación del laboratorio.

Una vez comprimidos en un único fichero ZIP **deberá adjuntarlo en la entrega del laboratorio Virtual a través de la Intranet – Solicitud de Docencia en Red**. Tras la entrega recibirá un acuse de recibo donde se le indicará una URL a la que tiene que acceder para rellenar un formulario web. **Sólo una vez que ha rellenado este formulario se podrá dar por entregado el laboratorio para su publicación en el servidor web.**

- Laboratorios que requieran la subida de archivos.

Los laboratorios que requieran la subida de **imágenes** por parte del usuario deben seguir las siguientes recomendaciones básicas de seguridad:

1. El botón de selección de imagen debe restringir la carga de archivos que no sean realmente imágenes.
2. La imagen cargada por el usuario no debe mostrarse directamente en pantalla. En su lugar, se debe mostrar una conversión de esta imagen. Sin embargo, para el procesamiento interno, se puede utilizar la imagen original.

Al programar la función que habilita la carga de imágenes, el laboratorio deberá corregir un error que introduce MATLAB al compilar el laboratorio. Un ejemplo simple de esto se presenta a continuación:

```
function AbrirficherodeimagenButtonPushed(app, event)

    [filename, pathname] = uigetfile({'*.png;*.jpg;*.jpeg', 'Image Files (*.png, *.jpg, *.jpeg)'}, 'Seleccione una imagen');

    if isequal(filename, 0)
        return;
    end

    [~, ~, ext] = fileparts(filename);
    if ~any(strcmpi(ext, {'png', 'jpg', 'jpeg'}))
        disp('formato incorrecto');
        delete(fullfile(pathname, filename));
        return;
    end

    fullpath = fullfile(pathname, filename);
    try
        img = imread(fullpath);
        app.foto.Value = fullpath;
    catch
        try
            disp('archivo de imagen corrupto');
            unscramble(fullfile(pathname, filename));
            app.foto.Value = fullpath;
            return;
        end
    end
end
```

Para que esto funcione, el laboratorio incluirá el archivo `unscramble.m`, que contendrá lo siguiente:

```
function unscramble(fname_in, fname_out)

% Read scrambled file
fin = fopen(fname_in, 'rb');
x_in = fread(fin);
fclose(fin);
```

```
% Preallocate output buffer for speed
fnd = find((x_in >= 194) & (x_in <= 195));
x_out = zeros(1,length(x_in)-length(fnd));

% Process scrambled file
i=1;
j=1;
while i <= length(x_in)
    switch x_in(i)
        case 194
            x_out(j) = x_in(i+1);
            i = i+1;
        case 195
            x_out(j) = x_in(i+1)+64;
            i = i+1;
        otherwise
            x_out(j) = x_in(i);
    end
    i = i + 1;
    j = j + 1;
end

% Write unscrambled file
fin = fopen(fname_out,'wb');
fwrite(fin,x_out);
fclose(fin);
```

Si se carga un **archivo distinto a una imagen**, también se deben aplicar las restricciones de seguridad. Por lo tanto, la función encargada de manejar la carga de archivos al servidor debe restringir el tipo de archivo y eliminarlo del servidor si no cumple con el formato especificado.

- **Laboratorios de ejemplo.**

Puede encontrar muchos ejemplos en RiuNet – Docencia – Objetos de aprendizaje – Plan Docencia en Red. Laboratorios virtuales: <http://riunet.upv.es/handle/10251/103850>

O puede revisar algunos de los laboratorios Matlab App Designer publicados:

- <https://laboratoriosvirtuales.upv.es/webapps/PulseGenerator.html>
- https://laboratoriosvirtuales.upv.es/webapps/Ana_loc.html

- **Uso de fórmulas.**

Los diferentes campos que componen el cuerpo de la página web que contiene el laboratorio (introducción, objetivo, instrucciones, etc.) permiten la inclusión de fórmulas en formato LaTeX encerradas entre `\(\displaystyle y \)`.



- **Dudas, problemas y consultas:** Enviar Grial a la aplicación “Laboratorios virtuales” o bien correo a docenciaenred@upvnet.upv.es