

**Materia: PROGRAMACIÓN III**

<b>Apellido:</b>		<b>Fecha:</b>	
<b>Nombre:</b>		<b>Docente<sup>(2)</sup>:</b>	
<b>División:</b>	<b>3C</b>	<b>Nota<sup>(2)</sup>:</b>	
<b>Legajo:</b>		<b>Firma<sup>(2)</sup>:</b>	
<b>Instancia<sup>(1)</sup>:</b>	<div style="display: flex; justify-content: space-around;"> <div>PP</div> <div></div> <div>RPP</div> <div>X</div> <div>SP</div> <div></div> <div>RSP</div> <div></div> <div>FIN</div> </div>		

Todo los archivos se deben tratar del mismo modo (serializado, JSON o línea a línea)

A partir del punto tres, todas las rutas deben estar autenticadas.

La aplicación no debe mostrar errores ni warnings.

Las respuestas deben ser en formato json.

1. **(POST) usuario.** Registrar a un cliente con email, clave y foto y guardarlo en el archivo users.xxx. Darle un nombre único a la imagen.
2. **(POST) login:** Recibe email y clave y si son correctos devuelve un JWT, de lo contrario informar lo sucedido. La clave no se debe guardar en texto plano.
3. **(POST) materia:** Recibe nombre, cuatrimestre y lo guarda en el archivo materias.xxx. Agregar un id único para cada materia.
4. **(POST) profesor:** Recibe nombre, legajo (validar que sea único) y lo guarda en el archivo profesores.xxx.
5. **(POST) asignacion:** Recibe legajo del profesor, id de la materia y turno (manana o noche) y lo guarda en el archivo materias-profesores. No se debe poder asignar el mismo legajo en el mismo turno y materia.
6. **(GET) materia:** Muestra un listado con todas las materias.
7. **(GET) profesor:** Muestra un listado con todas las profesores.
8. **(GET) asignacion:** Muestra un listado con todas las materias asignadas a cada profesor.
9. **(POST) usuario/:email:** Recibe una imagen y se la asigna al usuario indicado. Guardar la imagen anterior en la carpeta **backup**.