

**Materia: PROGRAMACIÓN III**

<b>Apellido:</b>		<b>Fecha:</b>	
<b>Nombre:</b>		<b>Docente<sup>(2)</sup>:</b>	
<b>División:</b>	<b>3D</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;"> <span>PP</span> <span>X</span> <span>RPP</span> <span></span> <span>SP</span> <span></span> <span>RSP</span> <span></span> </div>	<b>FIN</b>	

**ANTES DE COMENZAR EL PARCIAL CREAR UN REPOSITORIO Y COMPLETAR EL FORMULARIO.**

**La clave para generar el token debe ser: `pro3-parcial`.**

**El payload del token debe contener por lo menos los siguientes claims: `email y tipo`.**

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.

1. **(POST) usuario.** Registrar a un cliente con email, clave y tipo (encargado, cliente) y guardarlo en el archivo users.xxx.
2. **(POST) login:** Recibe email y clave y si son correctos devuelve un JWT, de lo contrario informar lo sucedido.
3. **(POST) pizzas:** (Solo para encargado). Recibe tipo (molde o piedra), precio, stock, sabor (jamón, napo, muzza) y foto y lo guarda en el archivo pizzas.xxx, a la imagen la guarda en la carpeta *imágenes*. Validar los tipos y sabores. La combinación tipo - sabor debe ser única.
4. **(GET) pizzas:** (Encargado y cliente) Muestra la lista de pizzas, si es encargado muestra el stock, si es user no muestra stock.
5. **(POST) Ventas:**(Solo clientes) Recibe tipo y sabor y si existe esa combinación y hay stock devuelve el monto total de la operación. Si se realiza la venta restar el stock a la pizza y guardar la venta archivo *ventas.xxx* el email, tipo, sabor, monto y fecha.
6. **(GET) ventas:** Si es encargado muestra el monto total y la cantidad de las ventas, si es cliente solo las compras de dicho usuario.
7. Generar una marca de agua al subir la foto en el punto 3.