

**Materia: PROGRAMACIÓN III**

<b>Apellido:</b>		<b>Fecha:</b>	
<b>Nombre:</b>		<b>Docente<sup>(2)</sup>:</b>	<b>Rampi-villegas</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></span> <span>RPP</span> <span>x</span> <span>SP</span> </div>	<div style="display: flex; justify-content: space-around;"> <span></span> <span>RSP</span> </div>	<div style="display: flex; justify-content: space-around;"> <span>FIN</span> </div>

Se debe realizar una API REST con slim framework.

Se deben respetar los nombres de los archivos y de las clases.

Se debe crear **una clase en PHP** por cada entidad y los archivos PHP solo deben llamar a métodos de las clases.

Todas las respuestas de la API deberán ser en formato JSON válido.

Las respuestas no deberán mostrar errores, warnings, var\_dump o cualquier otra cosa que no sea JSON.

1- (2 pts.) **Ruta: pizzas** (POST): se ingresa precio, Tipo ("molde" o "piedra"), cantidad( de unidades),sabor (muzza;jamón; especial), precio y dos imágenes (guardarlas en la carpeta images/pizzas y cambiarles el nombre para que sea único). Se guardan los datos en en el archivo de texto **Pizza.xxx**, tomando un id autoincremental como identificador, la combinación tipo - sabor debe ser única.

2- (2 pts.) **Ruta: pizzas**: (GET): Recibe Sabor y Tipo, si coincide con algún registro del archivo **Pizza.xxx**, retornar la cantidad de producto disponible, de lo contrario informar si no existe el tipo o el sabor. La consulta debe ser *case insensitive*.