

**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>	<b>PP</b>	<b>RPP</b>	<b>x</b>
		<b>SP</b>	
		<b>RSP</b>	
		<b>FIN</b>	

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 (muza;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*.

3- (1 pt.) A partir de este punto, se debe guardar en un archivo info.log la información de cada petición recibida por la API (ruta, metodo, hora).

4-(2 pts.) **Ruta: ventas** (POST). Recibe el email del usuario y el sabor,tipo y cantidad ,si el item existe en **Pizza.xxx**, y hay stock guardar en el archivo de texto **Venta.xxx** todos los datos , más el precio de la venta, un id autoincremental y descontar la cantidad vendida. Si no cumple las condiciones para realizar la venta, informar el motivo.

5- (2 pts.) **Ruta: pizzas** (PUT). Se reciben los datos a modificar, incluidas las imágenes.En el caso de haber imágenes, se deben mover las imágenes viejas a la carpeta images/backup.

6- (2 pts.) **Ruta: ventas** (GET). Debe recibir tipo y/o sabor y mostrar las coincidencias.

7- (2 pts.) **Ruta: pizzas**(DELETE), Recibe id de la pizza, de encontrarse en el archivo se deben borrar esos datos y mover la foto a la carpeta **/backUpFotos** y colocarle al nombre la fecha de hoy.

8- (1 pt) **Ruta: logs** (GET). Recibe una fecha y muestra los logs posteriores a esta.