Especialización en Back End III

# Procesando peticiones

##### Ejercitación grupal

##### Nivel de complejidad: medio 🔥🔥

# Problema

Un supermercado necesita un sistema para gestionar los productos frescos que tienen publicados en su página web. Para poder hacer esto, necesitan un **servidor** que ejecute una API que les permita manipular los productos cargados desde distintos clientes. Los campos que conforman un producto son:

| Nombre | Tipo de dato | Descripción | Ejemplo |
| --- | --- | --- |
| **id** | number | Identificador en conjunto de datos | **15** |
| **name** | string | Nombre caracterizado | **Cheese - St. Andre** |
| **quantity** | number | Cantidad almacenada | **60** |
| **code\_value** | string | Código alfanumérico característico | **S73191A** |
| **is\_published** | boolean | El producto se encuentra publicado o no | **True** |
| **expiration** | string | Fecha de vencimiento | **12/04/2022** |
| **price** | number | Precio del producto | **50.15** |

|  | **IMPORTANTE** |
| --- | --- |
|
| En el siguiente [enlace](https://drive.google.com/file/d/13KGYBrLPzVo_9YwpLw2jv6xaYp7C4Sg4/view), verás la solución del ejercicio anterior para que puedas continuar si no lograste terminar. | |

Ejemplo de payload

{

*"name"*:"Chicken",

*"quantity"*: 1,

*"code\_value"*:"AF429XD",

*"is\_published"*: true,

*"expiration"*:"10/11/2009",

*"price"*: 10.50

}

# 

# Ejercicio: Método PUT

Añadir el método [PUT](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/PUT) a nuestra API. Recordemos que este crea o reemplaza un recurso en su totalidad con el contenido en el request. Debemos validar los campos que se envían, tal como hicimos con el método POST. **Seguimos aplicando los cambios sobre la lista cargada en memoria**.