Especialización en Back End III

# Práctica clase 13

##### 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 de distintos clientes. Los campos que conforman un producto son:

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Tipo de dato  **JSON** | Tipo de dato  **Go** | Descripción | Ejemplo |
| **id** | number | int | Identificador en conjunto de datos | **15** |
| **name** | string | string | Nombre caracterizado | **Cheese - St. Andre** |
| **quantity** | number | int | Cantidad almacenada | **60** |
| **code\_value** | string | string | Código alfanumérico característico | **S73191A** |
| **is\_published** | boolean | bool | El producto se encuentra publicado o no | **True** |
| **expiration** | string | string | Fecha de vencimiento | **12/04/2022** |
| **price** | number | float64 | Precio del producto | **50.15** |

|  |  |
| --- | --- |
|  | **IMPORTANTE** |
|
| En el siguiente [enlace](https://drive.google.com/file/d/1_9GrRyr6goeRSfVGfMqnJX0gNZulsW8b/view), te dejamos la solución del ejercicio anterior para que puedas continuar si no lograste terminarlo. | |

# Ejercicio: Middleware - Response

De igual manera que implementamos un middleware para los **request**, vamos a implementar uno para los **response**. La función de esta herramienta es llevar un registro de las consultas realizadas, es decir, un **logger**. El paquete Gin utiliza uno por defecto, nosotros vamos a crear uno propio. Para sacarlo, reemplazamos:

r := gin.Default()

Por:

r := gin.New()

r.Use(gin.Recovery())

La función **gin.New()** retorna un motor sin ningún middleware adicional, luego **r.Use(gin.Recovery())**, añade un middleware para recuperar la ejecución del programa en caso de que ocurra un **panic()**. Debe llevar registro de:

* **Verbo utilizado**: GET, POST, PUT, etc.
* **Fecha y hora**: podemos utilizar el paquete [time](https://pkg.go.dev/time#Now).
* **URL de la consulta**: localhost:8080/products
* **Tamaño en bytes**: tamaño de la consulta.

