

Especialización en Back End III

Mesa de trabajo

- 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](#), te dejamos la solución del ejercicio anterior para que puedas continuar si no lograste terminarlo.

Ejercicio: Swagger.io

Documentamos nuestra API utilizando Swagger, pero haremos su [implementación en Go](#). Recuerden lo visto en clase, también les dejamos algunos tips extra:

- Para utilizar **swaggo** debemos [instalarlo](#) y, cada vez que busquemos actualizar la documentación, debemos correr el comando **swag init**.
- Debemos añadir a nuestro entorno la variable **HOST** con la dirección de nuestra API.
- La ruta de la path a la página de swagger es por defecto: **/swagger/index.html**.
- Un ejemplo de la documentación del método Post puede ser:

```
// Post godoc
// @Summary      Create a new product
// @Description  Create a new product in repository
// @Tags         products
// @Produce      json
// @Param        token header string true "token"
// @Param        body body domain.Product true "Product"
// @Success      201 {object} web.response
// @Failure      400 {object} web.errorResponse
// @Failure      404 {object} web.errorResponse
// @Router       /products [post]
func (h *productHandler) Post() gin.HandlerFunc {}
```