

**Aclaración: PDO: obligatorio.**

**POO: obligatorio el uso de POO. Todos los métodos deben estar declarados dentro de clases.**

**Base de Datos: crear y adjuntar el script de la base de datos (con sus tablas) para realizar la funcionalidad pedida.**

*Crear un **API Rest** para el super de Don Fu, que interactúe con la clase **Empleado** y **Producto** con la base de datos **DonFuBD** (Empleados - Productos). El objetivo es tener la siguiente funcionalidad:*

- 1.- (POST) Agrega un empleado a la BD (id-nombre-apellido-email-foto-legajo-clave-perfil)
- 2.- (POST) /email/clave/ Verifica empleado (email-clave). Retorna un json {valido:true/false, usuario:{datos}}
- 3.- (GET) Retorna un array de json conteniendo el listado completo de los empleados.
- 4.- (POST) /productos/ Agrega un producto a la base de datos (id-nombre-precio)
- 5.- (GET) /productos/ Retorna un array (json) de todos los productos.
- 6.- (PUT) /productos/ Modifica un producto
- 7.- (DELETE) /productos/ Elimina un producto

Crear los siguientes middleware que cumplan:

- 8.- Verificar empleado para POST, PUT y DELETE. Retorna un json {valido:true/false, usuario:{datos}}. Si es 'admin', acceso total. Si es 'user', solo post
- 9.- Guardar en archivo de texto la cantidad de veces que se accedió al api Rest (/productos/)
- 10.- Guardar en otro archivo de texto la fecha, hora, minutos y segundos más nombre y apellido de todos los que accedan a /productos/(post)
- 11.- Generar y guardar en la base de datos el JWT para aquellos usuarios cuya clave sea 1234.

**IMPORTANTE:**

***Se pueden bajar templates de internet o traer código hecho, pero en ningún caso se debe incluir código obsoleto o que no cumpla ninguna función dentro del parcial.***