**Reviewer de restaurantes🍝**

Hola mochileros, a continuación se les presenta un reto donde tendrán que trabajar de manera individual. En este reto podrás poner a prueba tus conocimientos sobre colecciones, patrones de diseño, pruebas unitarias…

Los **requerimientos técnicos** mínimos para cumplir este reto son los siguientes:

1. Diagrama de clases UML y diagrama relacional.
2. Implementación de 4 patrones de diseño:  
   Singleton - Para gestionar la instancia única de un repositorio central de datos (almacena restaurantes, menús y reviews).

Factory - Para crear instancias de diferentes tipos de reviews (reviews de restaurante y reviews de plato).

Observable - Para notificar automáticamente a los usuarios (u otras partes del sistema) cuando se agregue una nueva review o cambie la calificación promedio de un restaurante/plato.

Command - Para suplantar el uso de los switch.

1. Uso de Colecciones como Map, Set o LinkedList ( a parte del uso de List o ArrayLists).
2. Coverage de clase de 100% en tus pruebas unitarias.
3. Uso de controllers, repositories y services.

**Requerimientos Funcionales** que debería de cumplir tu app**:**

a) Gestión de Restaurantes

* Crear, editar y eliminar información de restaurantes.
* Consultar la lista de restaurantes disponibles.

b) Gestión de Menús

* Asociar un menú a un restaurante.
* Añadir, editar y eliminar platos en un menú.

c) Reviews de Restaurantes

* Crear reviews generales para un restaurante (calificación y comentario).
* Listar las reviews asociadas a un restaurante.
* Calcular la calificación promedio de un restaurante basada en las reviews.

d) Reviews de Platos

* Crear reviews para platos específicos (calificación y comentario).
* Listar las reviews de un plato.
* Calcular la calificación promedio de un plato.

Esta aplicación se manejara **POR CONSOLA**, no es necesario realizar un front para esta aplicación… aún.

**Recuerda que los repositorios que crees no pueden llevar los nombres ”Sofka”o “SofkaU”.**