



Examen práctico: Modelo 1

Nota previa

Junto con el resto del examen debes entregar la **List** desarrollada en prácticas, así como cualquier otra estructura de datos desarrollada en clase que hayas utilizado en el examen. Recuerda comprobar que el fichero que subas contiene todos los ficheros fuente necesarios y que descomprime correctamente.

EJERCICIO 1 (3.5 puntos)

En el código suministrado se ha implementado una versión mono-hilo de un restaurante (clase `CocinaMonohilo`). Analiza su comportamiento para implementar una versión multi-hilo del mismo (`CocinaMultihilo`). La implementación debe cumplir las siguientes condiciones:

- El número de `Cocineros` será un parámetro que le pasaremos durante la construcción del objeto.
- Debe haber paralelismo real entre los diferentes `Cocineros` de la `CocinaMultihilo`.
- En el `Horno` solamente puede cocinarse una `Pizza` simultáneamente.
- No es necesario conservar una correspondencia entre el orden de los pedidos y de las `Pizzas` resultado.
- Puedes modificar y crear las clases que consideres necesarias.

En el método `Ejercicio1` del proyecto `ejercicios` escribe un pequeño código de ejemplo y asegúrate de que el comportamiento es el solicitado.

EJERCICIO 2 (3.5 puntos)

Escribe consultas LINQ para contestar a las siguientes cuestiones:

- Obtén el nombre de las `Pizzas` del menú que no llevan tomate.
- Obtén los nombres de los `Ingredientes` con alérgenos que se utilizan en la elaboración de las `Pizzas` del menú

Se penalizará el uso de código imperativo innecesario.

Implementa ambas consultas en el método `Ejercicio2` del proyecto `ejercicios`.

EJERCICIO 3 (3 puntos)

A tu implementación de `List<T>` realizada en clase, debes añadir un método `DeleteIfFollowing`. Ese método recibe como parámetro un delegado de tipo `Func<T, T, bool>`, el cual se aplicará a cada elemento de la lista (excepto el último) y su sucesor. Si devuelve `true`, entonces se eliminará el sucesor y se volverá a probar con el nuevo sucesor. Este método **debe modificar la lista actual**.

Dada una lista de enteros, prueba su funcionamiento eliminando (`DeleteIfFollowing`) elementos sin estos son mayores que el actual o cualquier otro criterio que se te ocurra. Para ello escribe un fragmento de código en el método `Ejercicio3` del proyecto `ejercicios`.