

Diseñar e implementar en Ruby un programa que utilice el patrón concurrente *productor-consumidor* y la metodología de pruebas unitarias.

1. Implementar una clase **Cocinero**.

- Esta clase debe contar con métodos que indiquen el nivel de **experiencia** del trabajador, el número de **años** que ha trabajado y el número de platos que produce **contador**.
- También debe tener un método que incremente el número de años trabajados por el cocinero **uno_mas**. Además de incrementar el número de años trabajados, cada año el nivel de experiencia del cocinero ha de crecer en la cantidad que decida el programador, y después de un cierto número de años el cocinero debe jubilarse. Durante los primeros años el cocinero no debe elaborar platos (son sus años de aprendizaje) pero sí debería hacerlo una vez transcurridos un cierto número de ellos, también fijados por el programador. Se espera que un cocinero con más horas trabajadas produzca más que un cocinero novato. Por supuesto, se debería ser capaz de contar el número de platos que produce el cocinero cada año.
- Finalmente, se ha de implementar un método que permita degustar un plato **degustar_uno**. Su invocación reduce el contador en uno y devuelve al comensal una cadena que dice lo *delicioso* que es, o quizás sólo le dice que *no hay* más platos o que el cocinero está *jubilado*.

2. Implementar un *thread* para entrenar un cocinero.

3. Implementar un *thread* para degustar platos.

La salida puede ser algo similar a lo siguiente:

```
$ ruby cocineros.rb
Sorry, no dishes to tast
Dish taster waiting patiently...
Sorry, no dishes to tast
Dish taster waiting patiently...
The dish is delicious
Dish taster waiting patiently...
Sorry, the Chef is retired
```