

Caso 1 - Infraestructura Computacional

Juan Andrés Romero C - 202013449

Juan Alegría - 202011282

Nuestro programa se compone principalmente de 5 clases que se pueden apreciar en el siguiente diagrama de clases:

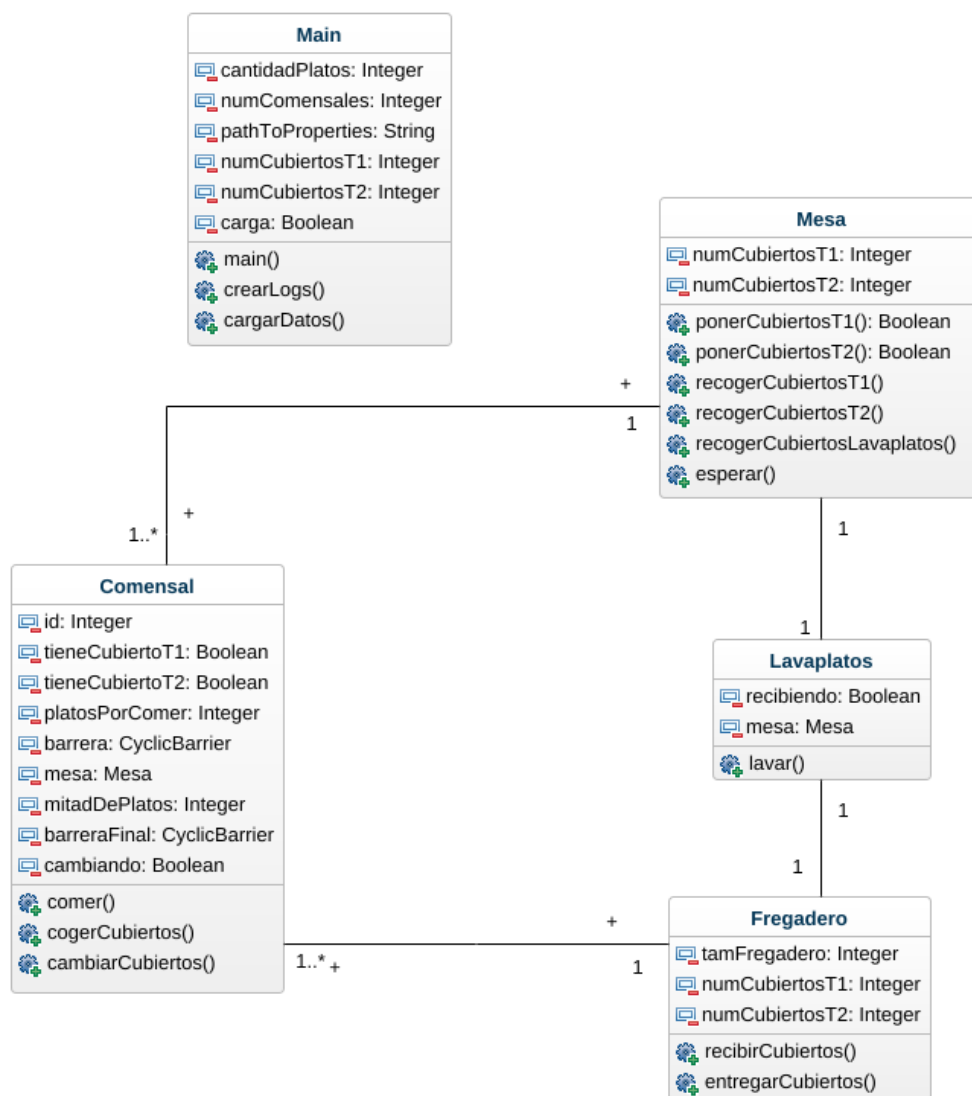


Fig 1. Diagrama UML del programa

Explicación general de las clases

- **Main:** Es la clase principal, donde se carga la configuración y se crean las instancias del resto de las clases. Representa la cena.
- **Mesa:** Es una clase estática que contiene la cantidad de cada tipo de cubiertos que están disponibles para ser usados por los comensales.
- **Comensal:** Es la clase que hace avanzar la cena al usar el método comer para hacer un progreso individual entre platos.
- **Lavaplatos:** Es la clase encargada de limpiar el fregadero y dejar en la mesa los cubiertos para disposición de los comensales.
- **Fregadero:** Es una clase estática que guarda la cantidad de cubiertos ya usados por los comensales. Tiene un máximo de cubiertos que puede guardar.

Relación y sincronización entre clases

- **Mesa y Comensal:** Un comensal recoge de la mesa un cubierto de tipo 1 y luego un cubierto de tipo 2. Esta acción está limitada por métodos de tipo “*synchronized*” para evitar que muchos comensales intenten coger cubiertos al mismo tiempo y haya una inconsistencia en la información de los cubiertos. El comensal está constantemente llamando al método cogerCubiertos hasta que sus dos atributos booleanos relacionados con los cubiertos estén en true. Primero se intenta coger un cubierto de tipo 1 y luego uno de tipo 2, si no hay suficientes cubiertos, el comensal se queda esperando hasta que haya disponibles. Si ya pudo coger uno de tipo 1, pero no hay de tipo 2, el comensal debe devolver el cubierto de tipo 1 y volver a intentar coger ambos. Adicionalmente, cuando la

mesa recibe cubiertos, esta le notifica a todos los comensales en espera, de que ya hay cubiertos disponibles.

- **Comensal y Fregadero:** Cuando un comensal acaba de comer un plato, éste deposita sus dos cubiertos al tiempo en el fregadero. Si no puede hacer esto por el límite de cubiertos en el fregadero, se queda esperando haciendo yield() hasta que pueda hacerlo.
- **Fregadero y Lavaplatos:** El lavaplatos recoge cubiertos del fregadero y los lava. Si no hay cubiertos disponibles para lavar, el lavaplatos se queda haciendo yield() hasta que estén cubiertos.
- **Lavaplatos y Mesa:** Después de lavar los cubiertos, el lavaplatos pone los cubiertos ya limpios en la mesa.

Funcionamiento del programa

Cuando se inicia el programa, la clase Main lee el archivo de configuración ("config.properties") e inicia una cena con los parámetros indicados. Todo lo que sucede en el programa es registrado en consola y en un archivo de logs, ubicado en la carpeta de logs del proyecto. Adicionalmente, para la validación del programa se usó el archivo de logs y lo impreso en consola, se revisaron estos logs manualmente para comprobar el correcto funcionamiento del programa.

Tipos de logs

- "Comensal " + id + " empezó a comer"
- "Comensal " + id + " empieza a comer. Platos por comer: " + platosPorComer
- "Comensal " + id + " llegó a la mitad de sus platos."
- "Todos los comensales terminaron de comer y se fueron para la casa."
- "Comensal " + id + " intenta coger un cubierto T1."
- "Comensal " + id + " no pudo coger un cubierto T1."

- "Comensal " + id + " intenta coger un cubierto T2."
- "Comensal " + id + " no pudo coger un cubierto T2."
- "Comensal " + id + " devolvió un cubierto T1."
- "Comensal " + id + " está esperando a que se desocupe el fregadero"
- "Comensal " + id + " cambió de cubiertos"
- "Se agregaron cubiertos al fregadero. El fregadero tiene: " + numCubiertosT1 + " cubiertos T1 y " + numCubiertosT2 + " cubiertos T2"
- "Se retiraron cubiertos del fregadero. El fregadero tiene: " + numCubiertosT1 + " cubiertos T1 y " + numCubiertosT2 + " cubiertos T2")
- "Lavaplatos recogió cubiertos del fregadero."
- "Lavando Cubiertos"
- "Ejecución de hora: " + LocalDateTime.now()
- "Se dejó un cubierto tipo T1 en la mesa. Cantidad disponible T1: " + numCubiertosT1 + " | Cantidad disponible T2: " + numCubiertosT2
- "Se dejó un cubierto tipo T2 en la mesa. Cantidad disponible T1: " + numCubiertosT1 + " | Cantidad disponible T2: " + numCubiertosT2
- "Cubiertos dejados en la mesa"
- "Cantidad de cubiertos en la mesa: T1 = " + numCubiertosT1 + " | T2 = " + numCubiertosT2
- "Lavaplatos está esperando que hayan cubiertos en el fregadero."

Notas adicionales

Al realizar la ejecución del programa con menos cantidad de tipo de cubiertos que comensales, y una cantidad diferente de tipo de cubiertos, puede llegar a darse el error de Stack Overflow. Esto se debe a la recursión relativamente infinita de un método en el comer, puesto que puede llegar a un punto en que en la mesa solo hay disponible un cubierto de un tipo y 0 del otro, entonces un comensal lo coge, pero no tiene el otro, entonces lo libera y avisa a todos, y otro lo coge y así sigue en un ciclo. De este modo, cambiamos la función recursiva para que el comensal reciba cubiertos a un ciclo while.