



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE INGENIERÍA



SISTEMAS OPERATIVOS

SEMESTRE 2025-2

JESÚS TALONIA FUENTES

PROYECTO 1

“Una situación cotidiana con concurrencia y sincronización”

La concurrencia y la sincronización son 2 conceptos muy relacionados entre sí, que son referentes a la programación paralela. Pero no se quedan sólo ahí; si se puede programar, es porque primero se pudo imaginar, y si se puede imaginar, por lo menos debe tener una base en el mundo real. Este proyecto consiste en eso, encontrar una aplicación a una problemática de la *vida real* y tratar de darle una solución usando hilos.

Planteamiento

Para este trabajo, se tomó de base la interacción entre 2 hermanos, roomies, ambos estudiantes de la unam, que al ser foráneos se ven forzados a rentar una habitación en la Ciudad de México, y lidiar con los quehaceres del “hogar” que ahora comparten. Las reglas del planteamiento son:

Los hermanos son Carlos y Jesús. Vienen a la ciudad de lunes a viernes para asistir a la universidad, y los fines de semana se regresan a su ranchito. Ambos tienen que ayudar con los quehaceres diarios para mantener limpio el espacio que comparten. Pero saben que, por X o Y razón, es posible que uno tenga más tiempo que el otro para realizar las tareas, así que en lugar de repartirlas equitativamente, si uno ve que hay cosas por hacer, las hace, pero no se sigue un orden en específico.

Las tareas a realizar son:

- Barrer
- Trapear (solo si primero se barrió)
- Sacar la basura
- Lavar los trastes sucios (como son foráneos, llevan contenedores con comida preparada para toda la semana)
- Comprar fruta (sólo se realiza los martes, aprovechando los martes de frescura en el supermercado)
- Rellenar el garrafón de agua (se usa un garrafón pequeño, de 10L, por lo que se debe rellenar todos los miércoles)

Además, como se trata de seres humanos, ambos hermanos necesitan comer y dormir.

Para comer se requiere tener cubiertos limpios. No se puede comer si éstos están sucios, cualquiera de los hermanos se debe asegurar de lavar sus cubiertos antes de comer. Al comer se vacía un contenedor de comida, que genera trastes sucios tanto por los contenedores como por los cubiertos, pero es posible que no se laven justo después de comer, que se laven hasta el día siguiente o que se laven por el hermano en el mismo día.

Para poder dormir, la lista de tareas del día debe estar vacía, y uno no se puede ir a dormir si el otro está haciendo *ruido* tratando de completar una tarea.

Carlos tiene un horario escolar muy bueno, no tiene clases los viernes, así que desde el jueves se regresa al ranchito, y ya no ayuda con las tareas de jueves y viernes. Esto significa que estos 2 días Jesús puede realizar las tareas en orden, para ser más eficiente con ellas. Pero también implica que (por lo menos el viernes) es completamente obligatorio lavar tanto los cubiertos como los contenedores sobrantes, para no dejar nada sucio y que el lunes *siempre* haya trastes limpios (ergo, los lunes no se lavan trastes, a menos de que se laven los cubiertos que se usen ese día).

¿Cómo resolver con hilos?

Es claro que cada roomie (hermano) representa un hilo diferente, que debe realizar tareas que son tanto pertenecientes a la lista compartida, como propias. Se debe cuidar la concurrencia sobre la lista de tareas compartidas, tanto al escoger una tarea para realizar, como al añadir tareas nuevas. Hay una tarea (trapear) que requiere de una tarea que la preceda, y para apagar las luces y continuar al siguiente día, es necesario sincronizar los hilos para que avancen al mismo tiempo (de lo contrario, es posible que un hilo quiera continuar su ciclo del día siguiente antes de que el otro termine el anterior).

El programa funciona de este modo:

Se utilizan las librerías `time`, `random` y `threading` para el correcto funcionamiento de los hilos. `pygame` y `sys` son usadas únicamente para la interfaz.

Se tiene una función `set_tareas()` que se encarga de rellenar la lista de tareas según el día de la semana. Para los casos de lunes, martes y miércoles hay tareas específicas que aquí se asignan, y mientras haya 2 roomies, se mezclan aleatoriamente las tareas para tener un poco de dinamismo en la ejecución.

Con las tareas inicializadas, se crean los hilos. Cada roomie tiene su propia función, que aunque son similares entre sí, tienen sus responsabilidades extra.

Todos los días los roomies despiertan y sienten hambre. Jesús lo hace por 5 días, Carlos solo por 3, para irse al cuarto. Usando un mutex que protege el acceso a las tareas compartidas, cada hilo toma una tarea de `stack_tareas` y la usa como argumento en la función `hacer_tarea(tarea, nombre)`. En esta, se hace un match con la duración de la tarea, y se marca como completada. Como ya dijimos, tienen que comer, por lo que hay un 33% de probabilidades de llamar a `comer()` para satisfacer su necesidad. Comer implica que haya cubiertos limpios. De ser así, se procede a comer, y al acabar hay un 50% de probabilidades de limpiar sus cubiertos, pero si no lo hace, tiene que reportar que ahora hay una tarea extra de lavar cubiertos. En ambos casos se generan también platos sucios por los contenedores de comida, por lo que en un mal día se pueden juntar platos sucios del día anterior, cubiertos sucios y contenedores del día.

Esto se repite en ciclo hasta que la lista de tareas se vacía. Si en todo ese rato no se comió, se debe comer antes de irse a dormir.

Como Jesús es el hermano mayor, su función es la que tiene más responsabilidades. Tiene que comprobar un semáforo que le señala si Carlos ha terminado con lo suyo. En tal caso, debe reportar que el día ha terminado en `logs[]`, avanzar la variable de los días transcurridos y, por último, establecer las tareas para el día siguiente.

Para irse a dormir, se usa un `rendezvous` entre los dos hilos. Se decidió así, porque las tareas extra de Jesús son esenciales, y por ello hay que asegurarnos de que Carlos no intente nada hasta que se hayan cumplido.

A partir del jueves, Jesús queda solo, y tiene que organizarse para no caer en un bloqueo si intenta esperar a Carlos para trapear o dormir (porque ya no está), y también adquiere la obligación de no dejar nada sucio los viernes. Para ello, la lista no se aleatoriza, y ya no checa el semáforo de trapear ni espera al `rendezvous` para dormir.

El resto de las instrucciones son propias de `pygame` para mostrar la interfaz gráfica.

Especificaciones y requerimientos

El programa fue creado con python 3.11 . Es necesario usar como mínimo la versión 3.10, pues se usa la estructura match - case que no está disponible en versiones anteriores.

Es necesario instalar la librería pygame, pues con ella se crea toda la interfaz gráfica del sistema. Las demás librerías que se importan ya vienen en la instalación por defecto, por lo que no hay que hacer mayores cambios.

Debido al uso de pygame, es necesario ejecutar el programa en un entorno con capacidades gráficas. Se probó su funcionamiento desde la terminal estándar de linux (Debian13), usando un IDE (VScode), y usando 2 compiladores en línea (Google Collab y onlineGDB). Los dos últimos no tienen capacidades gráficas (Collab requiere bastantes artimañas para engañar al código y correrlo, pero no es efectivo), por lo que el programa no hace nada.

Una vez compilado y ejecutado, se abre una pantalla con la simulación de los roomies. Se requiere pulsar [ENTER] para iniciar. En pantalla se mostrarán los logs (que sería la salida estándar en una terminal) junto con cuadrados de colores que representan las tareas y los roomies.



Los cuadros superiores son los roomies, señalados solo por la inicial de su nombre. Si el cuadrado es turquesa, el roomie está despierto y haciendo algo. Cuando se torna azul oscuro, es porque se fue a dormir. Para el caso de Carlos, cuando se va al ranchito su cuadrado se vuelve gris.

Los cuadrados inferiores son de las tareas. Si la tarea está pendiente, se ve en rojo, y cambia a verde cuando se completa. Si la tarea no está disponible para el día actual, se verá en gris.

Sólo se tiene que presionar la tecla [ESPACIO] al final de cada día (se indica en pantalla cuándo es esto) para continuar la simulación. Al llegar al día 5, se debe cerrar el programa pulsando [ENTER]

Simulación de Roomies

Hoy es Lunes

```
=== Día 1 ===
Tareas asignadas: ['Barrer', 'Sacar la basura', 'Trapear']
[Jesus] haciendo: Trapear
[Carlos] haciendo: Sacar la basura
[Carlos] quiere comer
[Carlos] Comiendo... Nam Nam
[Carlos] Lavando los cubiertos que usó
[Carlos] haciendo: Lavar los trastes
[Carlos] haciendo: Barrer
[Carlos] terminó sus tareas del día
[Jesus] quiere comer
[Jesus] Comiendo... Nam Nam
[Jesus] Dejó los cubiertos sucios :/
[Jesus] terminó sus tareas del día
>> Presiona ESPACIO para avanzar al siguiente día
```

J

C

Lavar

Barrer

Trapear

Sacar

Comprar

Llenar

Ejecución exitosa del día Lunes.

Simulación de Roomies

Hoy es Miercoles

```
Tareas asignadas: ['Trapear', 'Sacar la basura', 'Lavar los trastes', 'Barrer', 'Llenar el garrafon']
[Jesus] haciendo: Llenar el garrafon
[Carlos] haciendo: Barrer
[Carlos] haciendo: Lavar los trastes
[Jesus] haciendo: Sacar la basura
[Carlos] quiere comer
[Carlos] está lavando los cubiertos antes de comer
[Carlos] Comiendo... Nam Nam
[Jesus] quiere comer
[Jesus] Comiendo... Nam Nam
[Carlos] Dejó los cubiertos sucios :/
[Carlos] haciendo: Lavar los trastes
[Jesus] Dejó los cubiertos sucios :/
[Jesus] haciendo: Lavar los trastes
[Carlos] haciendo: Lavar los cubiertos
[Jesus] haciendo: Lavar los cubiertos
[Carlos] haciendo: Trapear
[Jesus] terminó sus tareas del día
[Carlos] terminó sus tareas del día
>> Presiona ESPACIO para avanzar al siguiente día
```

J

C

Lavar

Barrer

Trapear

Sacar

Comprar

Llenar

Ejecución exitosa del día Miércoles, Como ese día se rellena el garrafón, su cuadro deja de ser gris (se reiniciará al siguiente día)

Simulación de Roomies

Hoy es Jueves

```
=== Día 4 ===
Tareas asignadas: ['Trapear', 'Sacar la basura', 'Barrer', 'Lavar los trastes']
[Jesus] haciendo: Lavar los trastes
=== Carlos se va al rancho ===
[Jesus] haciendo: Barrer
[Jesus] haciendo: Sacar la basura
[Jesus] haciendo: Trapear
[Jesus] quiere comer
[Jesus] está lavando los cubiertos antes de comer
[Jesus] Comiendo... Nam Nam
[Jesus] Lavando los cubiertos que usó
[Jesus] haciendo: Lavar los trastes
[Jesus] terminó sus tareas del día
>> Presiona ESPACIO para avanzar al siguiente día
```

J

C

Lavar

Barrer

Trapear

Sacar

Comprar

Llenar

Ejecución exitosa de Jueves. Sólo se puede ver a Jesús haciendo tareas, porque Carlos ya se fue



Ejecución exitosa del día Viernes. Aquí la instrucción cambia, de pedir [ESPACIO] a pedir [ENTER] para concluir con la ejecución