

Integrantes: Méndez Sánchez Erick Jair

Rodriguez Colorado Carla Elizabeth

### **Proyecto 1: Situación cotidiana paralelizable**

#### **1. Descripción de los mecanismos de sincronización empleados.**

mutex: mecanismo de prevención que mantiene en espera a aquellos hilos que quieran entrar a una sección crítica del proceso hasta que el hilo que ejecuta esa sección termine.

Solo un hilo puede ingresar a la vez al segmento de proceso que es protegido por el mutex.

barrera: Generalización del rendezvous, tiene por objetivo frenar el avance en la ejecución de varios hilos hasta que todos alcancen un punto en específico. Permite la sincronización de varios hilos.

multiplex: Permite el paso de una cantidad específica de hilos a una región crítica dentro de un proceso.

#### **2. Lógica de operación.**

No se utilizó ninguna *variable compartida* a parte de los patrones, de estos se utilizaron:

repcionistas: multiplex que permite representar el número de recepcionistas disponibles para contestar los teléfonos

pizzeris: multiplex que permite representar el número de personas que disponibles en la producción de las pizzerias

caja\_Reparto: barrera que permite implementar que el repartidor del negocio únicamente pueda salir del local al repartir 3 órdenes

queso: mutex que representa el uso de ingredientes para la preparación de la pizza

horno: multiplex que representa la regla de negocio en donde solo tres pizzas podrán estar dentro para su cocción

#### *Descripción algorítmica por hilo/proceso*

Hilo principal

1. Entrada de n numero de clientes
2. Creación de n hilos pedido

Hilo pedido

1. Recepción del pedido
  - a. Agarrar un multiplex para indicar que una recepcionista está ocupada
  - b. Generación de número de pizzas al pedido
  - c. Liberar el multiplex de recepcionista
2. Producción
  - a. Adquirir un multiplex para indicar que un cocinero trabaja en el pedido
  - b. Creación de diferentes hilos (pizzas) para representar el avance de estas

- c. Guardar los hilos de las pizzas de la orden
- d. Esperar a que todas las pizzas sean sacadas del horno
- e. Se libera el cocinero a cargo de la orden

### 3. Reparto

- a. Pedido asignado al repartidor (Se abre un hilo)  
Una barrera global de 3 espera
- b. Si tiene menos de tres pedidos espera
- c. Si tiene tres órdenes sale al reparto

Hilo cocinar.

1. Se adquiere la zona de preparado con ingredientes
2. Se libera la zona de preparado
3. Si hay menos de tres pizzas en el horno
  - a. Se mete la pizza al horno
  - b. Se libera un lugar en el horno
  - c. Se pasa al paso 5
4. Si no hay menos de tres pizzas en el horno
  - a. Se espera a meter la pizza al horno
  - b. Se regresa al paso 3 cuando se es recordado
5. La pizza esta lista

*Descripción de la interacción entre los hilos.*

Entre el hilo principal y pedido: únicamente se inicia el hilo pedido.

Pedido y cocinar: El hilo a traves de join() espera que los hilos creados en cocina terminen para continuar.

Pedido y reparto: La inicialización de un hilo reparto en pedido indica que la orden está lista para su envío y se espera en barrera hasta que se cuente con los tres pedidos necesarios para ser liberado.

### **3. Descripción del entorno de desarrollo, suficiente para reproducir una ejecución exitosa**

*¿Qué lenguaje se emplea? ¿Qué versión?*

Se utilizó Python para el desarrollo de este proyecto, de forma específica sobre la versión de Python 3.7 proporcionada por Colab de Google

*¿Qué bibliotecas más allá de las estándar del lenguaje se utilizan?*

Solo se hizo uso de threading, random y time. Por lo que no se requiere de bibliotecas adicionales.

*¿Bajo qué sistema operativo/distribución lo desarrollaron y/o probaron?*

Fue desarrollado y probado tanto en Windows 10 como Windows 11

#### 4. Ejemplos o pantallazos de una ejecución exitosa

```
Ingresar el numero de clientes a atender 4
Buenas tardes, 0
Pedido no 0 recibido
Ok cliente 2 pizzas
Pizzera ocupada
Preparando la pizza 0
Buenas tardes, 1
Pedido no 1 recibido
Ok cliente 3 pizzas
En el horno 0
Preparada
Pizzera ocupada
Preparando la pizza 1
En el horno 1
Preparada
pizza 0 finalizadaPreparando la pizza 0
En el horno 0
PreparadaBuenas tardes, 2
```

```
pizza 1 finalizada
Pizzera desocupada
Pedido no 2 recibido
Ok cliente 3 pizzas
Pizzera ocupada
```

```
Repartidor esperandoPreparando la pizza 0
En el horno 0
Preparada
Buenas tardes, 3
Preparando la pizza 1
En el horno 1
Preparada
Preparando la pizza 1
Pedido no 3 recibido
Ok cliente 2 pizzas
```

```
En el horno 1
Preparada
Preparando la pizza 2
pizza 0 finalizadapizza 0 finalizada
pizza 1 finalizada
```

```
pizza 1 finalizada
En el horno 2
Preparada
pizza 2 finalizada
Pizzera desocupada
Pizzera ocupadaRepartidor esperando
Preparando la pizza 2
```

```
En el horno 2
PreparadaPreparando la pizza 0
En el horno 0
Preparada
```

Nota: En un inicio se solicita como dato de entrada el número de clientes que serán atendidos, se espera que se ingrese un valor entero positivo.