Universidad Nacional Autónoma de México

Facultad de Ingeniería

Ingeniería en Computación

Sistemas Operativos

Proyecto 2

Profesor: Gunnar Eyal Wolf Iszaevich

Alumnos:

Bárcenas Avelar, Jorge Octavio

Reza Chavarria, Sergio Gabriel

# Proyecto 2: Una situación cotidiana paralelizadle

# Identificación y Descripción del problema

En el caso a desarrollar se planteó la situación de un restaurante. El estado del restaurante consta de una cantidad de meseros, una cantidad de mesas, una cantidad de chefs, una fila de espera y clientes. La cantidad de mesas es mayor a la cantidad de meseros, por lo que los meseros deben de atender varias mesas. Los procesos o hilos planteados en el trabajo son los puestos de clientes (e invitados), meseros y chefs.

Los clientes son los que serán acomodados en la fila hasta que llegue su turno. El mesero llevará al cliente. El cliente llevará a los invitados. Los invitados son personas que no esperaron en la fila, sino ya venían con el cliente.

El mesero solo podrá tomar la orden cuando el cliente y los invitados ya decidieron que ordenar, para no tener varios viajes. El mesero llevará la orden al chef, los cuales se encargarán de realizar la orden, mientras que el mesero, si es necesario, atenderá más mesas. Cuando el chef tenga toda la orden completa, llamará al mesero para entregarla y continuará con las ordenes pendientes. Al entregarla a los comensales, estos tomarán su tiempo para la degustación de su comida. Cuando todos los comensales de la mesa terminen su comida, pedirán y pagarán la cuenta. Cuando terminen se liberará esta mesa para su uso. Todo el proceso se repetirá hasta que ya no haya comensales esperando en la fila.

Los casos de concurrencia se encuentran en los casos de los chefs y las ordenes, el chef debe de tomar su tiempo para preparar la orden, por lo que solo puede tomar una orden a la vez. Al finalizar y entregar la orden continuará con la siguiente orden pendiente. Además, que se encuentran varios chefs en la cocina, así se puede controlar la cantidad de ordenes pendientes.

# Mecanismos utilizados

Los mecanismos utilizados para la sincronización de procesos fueron

* Mutex: Utilizado para las funciones que tienen los meseros, y en la fila de clientes del restaurante.
* Semaforos: Uitlizado para los meseros disponibles, en las ordenes, en el estado del cliente ya comiendo.
* Multiplex: Utilizado para la cantidad de chefs, meseros, y mesas disponibles para ser utilizados.

En lenguaje de desarrollo utilizado fue Python (Versión 3.8.1). Ya que ya habíamos trabajado con anterioridad los conceptos de los sincronizadores de procesos. Se trabajaron en 2 sistema operativo.

* Linux Mint 19.1
* Windows 10