

Proyecto 1

Participantes:

- Vazquez Sanchez Erick Alejandro

Planteamiento del problema:

Puesto de tacos

En un puesto de tacos cotidiano de la CDMX todo mundo pide tacos al mismo tiempo y son atendidos en algunas ocasiones por un solo taquero.

Esta situación puede ser concurrente porque todos quieren pedir sus tacos sin importar que el taquero se encuentre atendiendo el pedido de alguien mas, a mi en lo personal me parece asombroso la habilidad del taquero para poder recordar cada uno de los pedidos que tiene, sin embargo por la saturación de trabajo puede llegar a olvidar alguno de los pedidos y tener un problema con el cliente.

Descripción de los mecanismos de sincronización empleados

Para la implementación de este problema se utilizó un semáforo que le dirá al *taquero* cuando ponerse a trabajar.

Para guardar los pedidos, se hizo uso de la *memoria* del taquero, la cual está protegida por un mutex que le permite hacer una nota mentalmente correcta sobre cada uno de los pedidos, además para implementar la saturación de pedidos mentales del taquero, se hace uso de un proceso en la *memoria*, esperando a que se tenga una cantidad x de pedidos mentales lo cual provocará que alguno se le olvide.

El *cliente* también tendrá un semáforo que le dirá cuando su orden esté lista, al suceder esto terminará su finalización, sin embargo si pasa mucho tiempo esperando, su enojometro aumentará y ya no va a querer los tacos.

Un punto importante a recalcar es que dos de los procesos, precisamente el que mantiene al taquero trabajando y el que olvida los pedidos se mantienen activos y no se llegan a concluir, por lo que mi taquero se queda en continua espera a nuevos pedidos. :c

Entorno y dependencias:

Para la implementación del proyecto se hizo uso de Python 3.8.10 junto a las librerías threading, time, random y tkinter. tkinter se utilizó para realizar una interfaz gráfica que solo sirve para insertar de manera cómoda el número de clientes que le van a llegar al taquero, pero el comportamiento del programa se observa en consola.

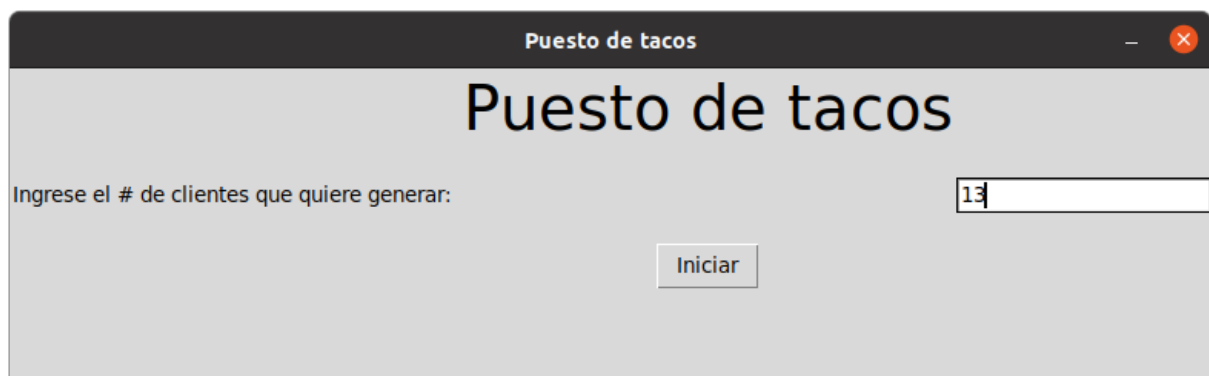
Ejecución:

Para la ejecución del programa se hace uso del siguiente comando:

`python3 app.py`

La ventana de la interfaz gráfica se cerrará una vez haya recibido los datos que pide, sin embargo como ya se mencionó existen dos procesos que se quedan activos por lo que para finalizar el programa se debe hacer uso de un ***CONTROL + C***

Capturas de pantalla de la ejecución del programa:



```
python3 app.py
KeyboardInterrupt:
→ VazquezErick git:(main) python3 app.py
  C5: Esperando pedido
T: Trabajando en el pedido del Cliente 5
  C7: Esperando pedido
  C0: Esperando pedido
  C1: Esperando pedido
  C3: Esperando pedido
  C10: Esperando pedido
  C4: Esperando pedido
  M : Olvidando pedido del cliente: 3
  C11: Esperando pedido
  M : Olvidando pedido del cliente: 1
T: Terminando una orden de  CAMPECHANO del Cliente 5
T: Trabajando en el pedido del Cliente 7
  C5: Gracias :D
  C9: Esperando pedido
  C6: Esperando pedido
  M : Olvidando pedido del cliente: 6
  C2: Esperando pedido
  M : Olvidando pedido del cliente: 11
  C8: Esperando pedido
  M : Olvidando pedido del cliente: 4
  C12: Esperando pedido
  M : Olvidando pedido del cliente: 12
T: Terminando una orden de  SUADERO del Cliente 7
T: Trabajando en el pedido del Cliente 0
  C7: Gracias :D
T: Terminando una orden de  CAMPECHANO del Cliente 0
T: Trabajando en el pedido del Cliente 10
  C0: Gracias :D
T: Terminando una orden de  PASTOR del Cliente 10
T: Trabajando en el pedido del Cliente 9
  C10: Gracias :D
T: Terminando una orden de  SUADERO del Cliente 9
T: Trabajando en el pedido del Cliente 2
  C9: Gracias :D
T: Terminando una orden de  PASTOR del Cliente 2
T: Trabajando en el pedido del Cliente 8
  C2: Gracias :D
T: Terminando una orden de  CAMPECHANO del Cliente 8
  C8: Gracias :D
  C12: ya no quiero mis tacos, ya me voy
  C4: ya no quiero mis tacos, ya me voy
  C6: ya no quiero mis tacos, ya me voy
  C1: ya no quiero mis tacos, ya me voy
  C11: ya no quiero mis tacos, ya me voy
  C3: ya no quiero mis tacos, ya me voy
```