



**Universidad Nacional Autónoma de México**

**Sistemas Operativos**

**Proyecto: Una situación cotidiana paralelizable**

**Miguel Adán Vargas López**

**Fecha de entrega: 6 de noviembre del 2023**

## **Planteamiento del problema**

Yo como soy de Torreón, Coahuila recuerdo con cariño las tradicionales gorditas de harina y de maíz de ahí, en mi ciudad hay puestos de gorditas en todos lados y hay de varios tipos, con mi programa busqué simular el operamamiento de uno de estos establecimientos, en especial de uno que va vender, gorditas de harina o maíz, gorditas al carbón y gorditas de cocedor, en el mismo local hay estos tres tipos de gorditas a la venta, considerando que las de harina y maíz son hechas por los mismos empleados pero son diferentes a los que trabajan las de carbón y las de cocedor.

En total, consideraré cuatro empleados:

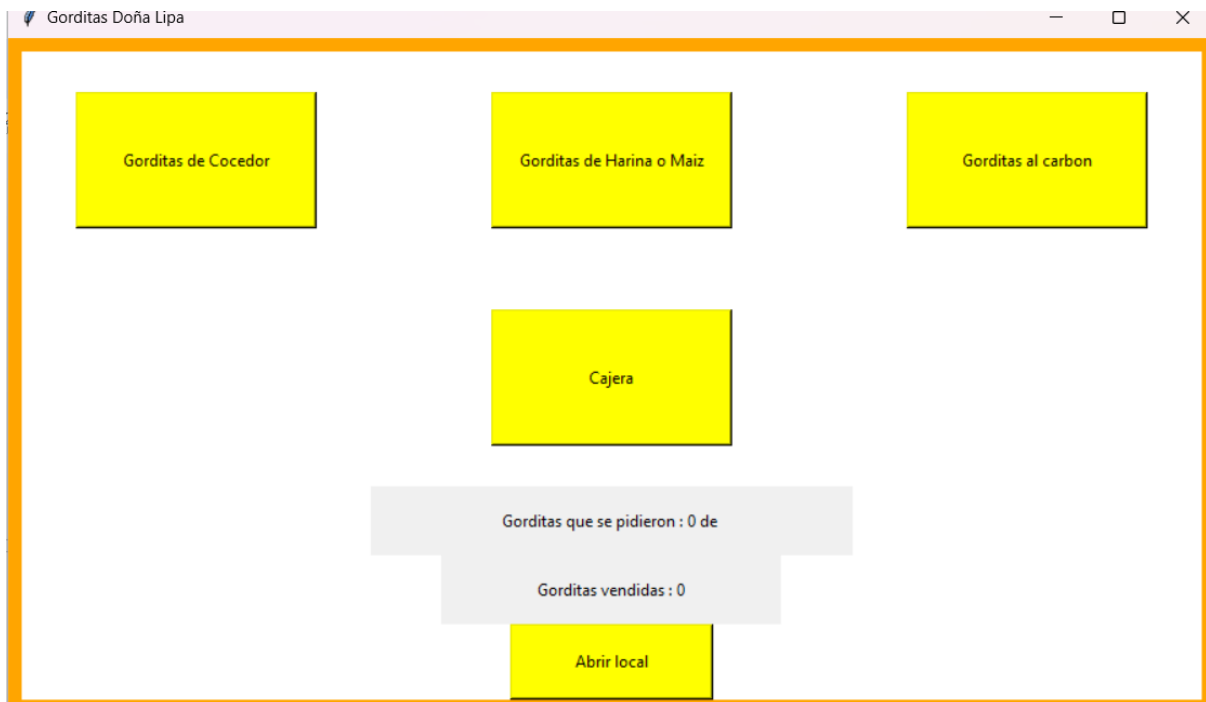
- Empleado en Caja
- Empleado que trabaja gorditas al carbon
- Empleado que trabaja gorditas de cocedor
- Empleado que trabaja gorditas de harina y maiz

La meta de mi proyecto es que el empleado de caja reciba pedidos y le pase una cantidad de gorditas a elaborar a cada una de los otros tres empleados y que ellos se dediquen a hacerlas cada quien en su puesto, por lo cual estos 3 deben tener la capacidad de trabajar simultáneamente y a su vez, el empleado de caja puede seguir recibiendo pedidos.

## **Diseño de interfaz**

En la interfaz desarrollada hay en total 6 widgets para los cuales importe Tkinter y 4 son botones y 2 son Labels, los botones uno es para la cajera, este botón lo que debe hacer es pedir un número aleatorio entre 1 y 4 a cada uno de los tipos de posibles gorditas, por otro lado, los otros tres botones deben agregar a su lista de espera una o dos gorditas de su tipo, eligiendo este valor también de forma aleatoria, mientras que los Label utilizados serán para mostrar información, como cuantas gorditas se pidieron y de que en el Label superior mientras que en el inferior, se muestra un total de cuantas gorditas se han hecho en total.

Creando la ventana principal seguido de los frames contenidos, donde uno funciona como un marco de color rojo y uno en el cual estan contenidos todos los widgets.



Para insertar los botones utilizamos un for anidado de otro para colocarlos según sus posiciones y a su vez, definir el texto que deben de llevar.

```

12 #creacion de ventana principal
13 root = Tk()
14
15 #Titulo de ventana con nombre del local de las gorditas
16 root.title("Gorditas Doña Lipa")
17
18 # Crear un frame para el borde del local
19 frame = Frame(root,bg="red")
20 frame.grid(row=2, column=0, rowspan=2, columnspan=3, sticky="nsew")
21 # Se crea el interior del local
22 frame_botones = Frame(frame,bg="white")
23 frame_botones.grid(row=2, column=0, rowspan=2, columnspan=3, sticky="nsew",padx=10,pady=10)
24
25 # Arreglo que almacenara los 4 botones de tipos de gorditas y la cajera
26 puestos = {}
27 #variable auxiliar para representar el texto que tendra cada boton
28 texto = ""
29
30 for fila in range(2):
31     for columna in range(3):
32         if columna == 0:
33             texto = f"Gorditas de Cocedor"
34         elif columna == 1 and fila == 1:
35             texto = f"Cajera, da click y toma ordenes"
36         elif columna == 1:
37             texto = f"Gorditas de Harina y Maiz"
38         else:
39             texto = f"Gorditas al carbon"
40         if fila == 0 or (fila == 1 and columna == 1):
41             btn_nuevo = Button(
42                 frame_botones,
43                 text=texto,
44                 justify="center",
45                 width=24,
46                 height=6,
47                 bg="yellow",
48                 fg="black",
49                 state="normal"

```

## Entorno utilizado

Desarrollado y probado en Windows 11 Home Single Language

Versión de Python con el que fue desarrollado el proyecto: 3.12.0

Se importaron librerías como Time, Threading y Tkinter para la interfaz gráfica.

## Desarrollo del programa

Mi objetivo con este programa es tener al menos tres hilos que representan cada empleado o grupo de empleados que trabajan haciendo los diferentes tipos de gorditas, que además pueden tardar más o menos para entregar los pedidos además del grupo principal que va ir lanzando pedidos a cada grupo de empleados. Para la toma de ordenes mi intención fue utilizar un Mutex para ayudar al acceso a las variables variables como pedidosCocedor por ejemplo, primero con un sleep se espera un nuevo pedido, y con valores aleatorios se define cuantas y de qué tipo de gorditas sera el pedido y se asigna a las variables globales.

```
49 def tomaOrdenes():
50     print("tomando ordenes")
51     #se deben tomar pedidos cada cierto tiempo aleatorio
52     sleep(random.random() * random.randint(10,20))
53     #segun el numero son la cantidad de gorditas que tienen que realizar
54     mutexTomaPedidos.acquire()
55     pedido = random.randint(1,4)
56     tipo = random.randint(0,2)
57     if tipo == 0:
58         pedidosCocedor += pedido
59         pedido_txt = "Se pidieron " + pedido + " de cocedor"
60         print(pedido_txt.text)
61     elif tipo == 1:
62         pedidosComal += pedido
63         pedido_txt = "Se pidieron " + pedido + " de maiz o harina"
64         print(pedido_txt.text)
65     else:
66         pedidosCarbon += pedido
67         pedido_txt = "Se pidieron " + pedido + " al carbon"
68         print(pedido_txt.text)
69     mutexTomaPedidos.release()
70
```

Los hilos tienen de propósito trabajar como un grupo de empleados separados e ir aliviando la demanda de los pedidos, los cuales tomarán cierto tiempo aleatorio y en el caso de las de comedor la idea es que se represente que son más tardadas que las otras dos y con la ayuda de un while tener funciones iterativas que no terminarán hasta acabar con la demanda de pedidos.

```
75 def cocinarCocedor(pedidos:int):
76     while (pedidos > 0):
77         sleep(random.random() * random.randint(5,9))
78         print("cocinando gorditas de cocedor")
79         pedidos-=1
80         cocinarCocedor(pedidos)
81         mutex.acquire()
82         contadorTotal+=1
83         mutex.release()
84 def cocinarCarbon(pedidos:int):
85     while (pedidos > 0):
86         sleep(random.random() * random.randint(3,6))
87         print("cocinando gorditas de carbon")
88         pedidos-=1
89         cocinarCarbon(pedidos)
90         mutex.acquire()
91         contadorTotal+=1
92         mutex.release()
93 def cocinarComal(pedidos:int):
94     while (pedidos > 0):
95         sleep(random.random() * random.randint(3,6))
96         print("cocinando gorditas de comal")
97         pedidos-=1
98         cocinarComal(pedidos)
99         mutex.acquire()
100        contadorTotal+=1
101        mutex.release()
```