



Proyecto 02

Simulacion Restaurante

Quintanar Ramírez Luis Enrique

1. Identificación y descripción del problema

Se planeó simular el funcionamiento de un restaurante, aunque no se hizo un restaurante completo (faltaron los cocineros), la concurrencia y el paralelismo se hicieron presente al tener clientes y meseros.

1. ¿Qué buscamos controlar?

Necesitamos asignar lugares a los clientes que llegan, pero no siempre van a existir los lugares necesarios, por lo que, si no hay cupo, los clientes decidirán si se van o se quedan.

Además, cada mesero deberá tomar las ordenes de los clientes a los que les dieron asientos, es normal en un restaurante que te atienda la misma persona que te llevó a tu mesa. Los meseros también pueden estar atendiendo a otras personas.

Los clientes se tardan un poco al elegir que van a comer así como se toman su tiempo para disfrutar sus alimentos

2. ¿Dónde pueden verse las consecuencias nocivas de la concurrencia?

Si han visitado un restaurante con mucha afluencia se darán cuenta que no es tarea fácil para los meseros mantener un orden, y en muchas ocasiones se ven superados por las circunstancias. Vamos, ¡quién quiere esperar cuando está hambriento!

La sociedad está acostumbrada a hacer filas para ingresar a lugares concurridos, donde no hay espacio para ellos, o están conscientes que deben esperar a ser atendidos, ¿qué pasaría si eso no ocurriera?. Lo hemos visto en situaciones de emergencia y pánico y digamos que... no nos gustaría estar siempre así

2. Descripción de los mecanismos de sincronización empleados

Para dar solución a estos problemas, únicamente utilizamos semáforos, uno para alertar al mesero de que no debe descansar en ese momento y que su cliente necesita algo, y otro para evitar que los clientes se "metan" en la fila y mantener un orden

3. Lógica de operación

1. Identificación del estado compartido(variables o estructuras globales)
Cuando un cliente llegó y ya sea que le den su asiento o que quiera esperar en la fila, se creará un objeto cliente, y para crearlo, los procesos no deben interferir uno con otro. Cuando a los clientes se les asigne un lugar, adentro o en la fila, se atiende al que llegue primero y los hilos no deben interferir cuando se los clientes se formen.
Los meseros no deben atender a la misma persona, por lo que tampoco pueden estorbarse entre ellos
2. Descripción algorítmica de cada hilo/proceso
Para los clientes:
Los clientes llegan según una probabilidad de llegada dada por el usuario, cuando llega, se crea un objeto Cliente, y alerta al mesero de que llegó, si tiene un asiento, ordena, come y se va, si no hay asiento decide si se queda o no, y hace fila según su decisión
Para los meseros:
Cuando son alertados de que llegó un cliente, se crea un objeto Mesero, el cual le dará asiento al cliente, le tomará la orden y despedirá al cliente
3. Descripción de la interacción entre ellos
Cuando el usuario ingresa los valores con los que se manejará la simulación, comienza la magia y los hilos se crean para empezar a interactuar. Primero llega un cliente que alertará a algún mesero de su llegada, el cual le dará su lugar ya sea un asiento en el restaurante o un espacio en la fila, mientras tanto, pueden seguir llegando más clientes. Cuando un mesero atiende a un cliente, espera a que tome su orden, y una vez que la tomó, el cliente disfruta de su comida y al terminar, alerta al mesero para que le de su cuenta y pueda marcharse.
Cuando el usuario decida terminar la simulación los hilos se detendrán y se terminará con la simulación

4. Descripción del entorno, suficiente para reproducir una ejecución exitosa

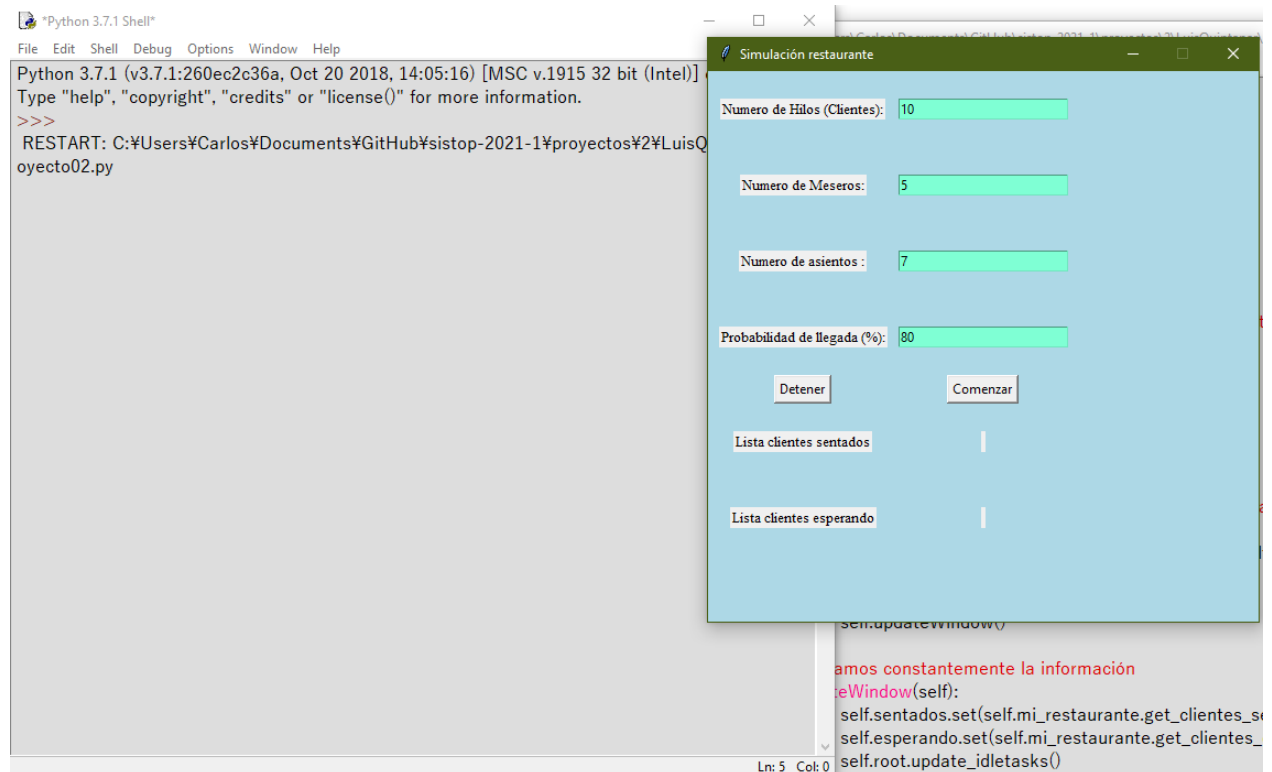
1. ¿Qué lenguaje emplean? ¿Qué versión?
Fue programado en lenguaje python, como es necesario tkinter, se recomienda usar de la versión 2 de python en adelante
2. ¿Qué bibliotecas van más allá de las estándar del lenguaje? Viene pre-instalado en python 3, en caso de no contar con ello, se deberá instalar
Para Fedora: \$ sudo dnf install python3-tkinter Para Ubuntu y derivados: \$ sudo apt-get install python3-tk Para Windows, en el shell de python: \$ pip install python-tk
3. ¿Bajo qué sistema operativo/distribución lo desarrollaron y/o probaron?
Fue desarrollo en dos sistemas operativos, en Windows 10 y Linux en la distribución de Ubuntu. En ambos sistemas fueron exitosas las ejecuciones, aunque con cambios en

su comportamiento.

En Windows cuando llega un cliente y quiere quedarse, lo pone en la lista de espera, cuando un cliente se marcha inmediatamente le da el asiento a un cliente en la fila. En cambio, en Linux, espera hasta que todos los clientes sentados sean atendidos para después dar asientos a los clientes que espera, teniendo en cuenta los clientes que van llegando mientras todas esos cambios ocurren.

5. Ejemplos de la ejecucion

En Windows:



```

Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Clientes sentados:Mesero 01 -atendiendo a 20
-
Clientes esperando: 3 -
Clientes sentados: 0 -2 -3 -
Clientes esperando:
Clientes sentados:Mesero 01 -atendiendo a 20
-3 -
Clientes esperando: 1 -
Clientes sentados: 0 -2 -3 -1 -
Clientes esperando:
Clientes sentados:Mesero 01 -atendiendo a 20
-3 -1 -
Clientes esperando: 4 -
Clientes sentados: 0 -2 -3 -1 Llego el cliente %- 24
-
Clientes esperando:
Clientes sentados:Mesero 01 -atendiendo a 20
-3 -1 -4 -Llego el cliente %
Clientes esperando: 3
9 -
Clientes sentados: 0 -2 -3 Llego el cliente %- 01
-4 -9 -
Clientes esperando:
Clientes sentados:Mesero 01 -atendiendo a 20
-3 -1 -4 -9 -
Clientes esperando: 7 -
Clientes sentados: 0 -2 -3 -1 -4 -9 -7 -Llego el cliente %
Clientes esperando: 1

MeseroClientes sentados: 10 atendiendo a - 02

```

```

projecto02.py - C:\Users\Carlos\Documents\GitHub\sisstop-2021-1\proyectos\2\LuisQuintanar\
Simulación restaurante
Numero de Hilos (Clientes): 10
Numero de Meseros: 5
Numero de asientos : 7
Probabilidad de llegada (%): 80
Detener Comenzar
Lista clientes sentados 0 2 3 1 4 9 7
Lista clientes esperando 6

#Actualizamos constantemente la información
def updateWindow(self):
    self.sentados.set(self.mi_restaurante.get_clientes_s
    self.esperando.set(self.mi_restaurante.get_clientes_
    self.root.update_idletasks()
    self.root.after(2 self undateWindow)

```

```

Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Tomando orden del clienteMesero 21
atendiendo a 2
Llego el cliente % 5
Llego el cliente % 3
Llego el cliente % 6
EL CLIENTE 2 YA NO TIENE MÁS QUE ORDENAR
..... Cliente 2 comiendo
.....EL cliente 2 se marcha
Clientes sentados:7 -6 -3 -5 -
Clientes esperando: 8 -
Clientes sentados: 7 -6 -3 -5 -8 -
Clientes esperando:
Tomando orden del clienteMesero 76
EL CLIENTE atendiendo a 77
YA NO TIENE MÁS QUE ORDENAR
..... Cliente 7 comiendo
Llego el cliente % 8
.....EL cliente 7 se marcha
Tomando orden del cliente 6
EL CLIENTE 6 YA NO TIENE MÁS QUE ORDENAR
..... Cliente 6 comiendo
.....EL cliente 6 se marcha
Clientes sentados:3 -5 -8 -
Clientes esperando: 4 -
Clientes sentados: 3 -5 -8 -4 -
Clientes esperando:
Clientes sentados:Mesero 30 -atendiendo a 53
-8 -4 -
Clientes esperando: 0 -
Clientes sentados: 3 -5 -

```

```

Simulación restaurante
Numero de Hilos (Clientes): 10
Numero de Meseros: 1
Numero de asientos : 5
Probabilidad de llegada (%): 90
Detener Comenzar
Lista clientes sentados 3 5 8 4 0
Lista clientes esperando

```

```

eWindow(self):
self.sentados.set(self.mi_restaurante.get_cli
self.esperando.set(self.mi_restaurante.get_c
self.root.update_idletasks()
self.root.after(2 self undateWindow)

```

The image shows a Python 3.7.1 Shell window and a 'Simulación restaurante' GUI window. The shell window displays the following log:

```

Clientes esperando: 10 -16 Llego el cliente %- 6917
-32 -
MeseroClientes sentados: 204 atendiendo a - 41
-3 -
Clientes esperando: 10 -16 -17 -32 -65 -
El cliente 65 se marchó
Clientes sentados: 4 -1 -3 -
Clientes esperando: 10 -16 -17 -32 -
MeseroClientes sentados: 204 atendiendo a - 41
-3 -
Clientes esperando: 10 -16 -17 -32 -59 -
El cliente 59 se marchó
Clientes sentados: 4 -1 -3 -
Clientes esperando: 10 -16 -17 -32 -Llego el cliente %
Clientes sentados:Mesero65
420 -atendiendo a 14
-3 -
Clientes esperando: 10 -16 -17 -32 -81 -
Clientes sentados: 4 -1 -3 -
Clientes esperando: 10 -16 -Llego el cliente % 1732
-32 -81 -
MeseroClientes sentados: 204 atendiendo a - 41
-3 -
Clientes esperando: 10 -16 -17 -32 -81 -53 -
Clientes sentados: 4 -1 -3 -
Clientes esperando: 10 -16 Llego el cliente %- 5917
-32 -81 -53 -
Clientes sentados:Mesero 420 -atendiendo a 14
-3 -
Clientes esperando: 10 -16 -17 -32 -81

```

The 'Simulación restaurante' GUI window displays the following parameters and lists:

- Numero de Hilos (Clientes): 100
- Numero de Meseros: 2
- Numero de asientos: 3
- Probabilidad de llegada (%): 90
- Detener (button)
- Comenzar (button)
- Lista clientes sentados: 4 1 3
- Lista clientes esperando: 10 16 17 32 81 53 88

The shell window also shows the following log:

```

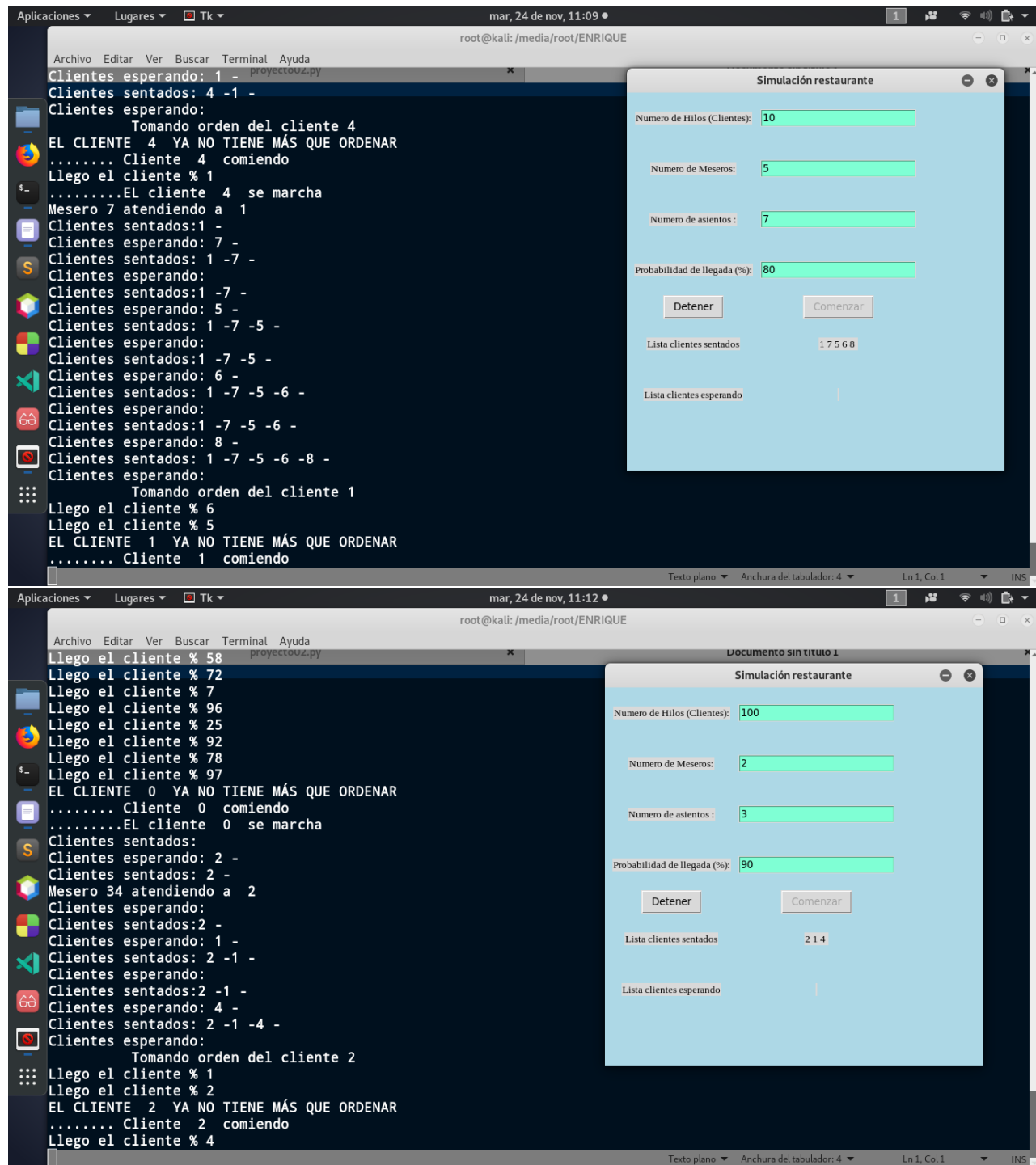
Clientes sentados: 4 -1 -3 -
Clientes esperando: 10 -16 -17 -32 -81 -53 -40 -41 -71 -64 -60 -86 -45 -72 -
Clientes sentados:Mesero 420 -atendiendo a 14
-3 -
Clientes esperando: 10 -16 -17 -32 -81 -53 -40 -41 -71 -64 -60 -Llego el cliente %
45 -72 -43 -
El cliente 43 se marchó
Clientes sentados: 4 -1 -3 -
Clientes esperando: 10 -16 -17 -32 -81 -53 -40 -41 -71 -64 -60 -86 -45 -72 -
Clientes sentados:Mesero 420 -atendiendo a 14
-3 -
Clientes esperando: 10 -16 -17 -32 -81 -53 -40 -41 -71 -64 -60 -86 -45 -72 -50 -
El cliente 56 se marchó
Clientes sentados: 4 -1 -3 -
Clientes esperando: 10 -16 -17 -32 -81 -53 -40 -41 -71 -64 -60 -86 -Llego el cliente %
-72 -
Clientes sentados:Mesero 420 -atendiendo a 14
-3 -
Clientes esperando: 10 -16 -17 -32 -81 -53 -40 -41 -71 -64 -60 -86 -45 -72 -60 -
El cliente 61 se marchó
Clientes sentados: 4 -1 -3 -
Clientes esperando: 10 -16 -17 -32 -81 -53 -40 -41 -71 -64 -60 -86 -45 -72 -
Clientes sentados:Mesero 420 -atendiendo a 14
-Llego el cliente % 356
-
Clientes esperando: 10 -16 -17 -32 -81 -53 -40 -41 -71 -64 -60 Llego el cliente %
-45 -72 -70 -
El cliente 70 se marchó
Clientes sentados: 4 -1 -3 -
Clientes esperando: 10 -16 -17

```

The GUI window also displays the following parameters and lists:

- Numero de Hilos (Clientes): 100
- Numero de Meseros: 2
- Numero de asientos: 3
- Probabilidad de llegada (%): 90
- Detener (button)
- Comenzar (button)
- Lista clientes sentados: 4 1 3
- Lista clientes esperando: 10 16 17 32 81 53 40 41 71 64 60 86 45 72

Para Linux:



The image displays two screenshots of a restaurant simulation application. The top screenshot shows the application running in a terminal window with a GUI overlay. The terminal output shows the simulation progress for client 64, including arrival, ordering, and departure. The GUI overlay, titled 'Simulación restaurante', shows input fields for the number of clients (100), waiters (2), and seats (3), along with a probability of arrival (90%). It also displays the current state of the simulation, including the list of seated clients (75 46 0) and the list of waiting clients (empty).

The bottom screenshot shows the application running in a terminal window with a GUI overlay. The terminal output shows the simulation progress for client 0, including arrival, ordering, and departure. The GUI overlay, titled 'Simulación restaurante', shows input fields for the number of clients (100), waiters (2), and seats (3), along with a probability of arrival (90%). It also displays the current state of the simulation, including the list of seated clients (77 44 15) and the list of waiting clients (16 37).

The image displays two screenshots of a restaurant simulation program. The top screenshot shows the initial state of the simulation, and the bottom screenshot shows the state after several customers have been served.

Top Screenshot:

- Terminal Output:**

```

.....EL cliente 3 se marcha
Mesero 9 atendiendo a 5
Clientes sentados:5 -4 -7 -1 -
Clientes esperando: 9 -
Clientes sentados: 5 -4 -7 -1 -9 -
Clientes esperando:
Clientes sentados:5 -4 -7 -1 -9 -
Clientes esperando: 0 -
Clientes sentados: 5 -4 -7 -1 -9 -0 -
Clientes esperando:
Clientes sentados:5 -4 -7 -1 -9 -0 -
Clientes esperando: 8 -
Clientes sentados: 5 -4 -7 -1 -9 -0 -8 -
Clientes esperando:
Tomando orden del cliente 5
EL CLIENTE 5 YA NO TIENE MÁS QUE ORDENAR
..... Cliente 5 comiendo
Llego el cliente % 8
Llego el cliente % 9
.....EL cliente 5 se marcha
Mesero 7 atendiendo a 4
Tomando orden del cliente 4
Llego el cliente % 0
EL CLIENTE 4 YA NO TIENE MÁS QUE ORDENAR
..... Cliente 4 comiendo
.....EL cliente 4 se marcha
Mesero 6 atendiendo a 7
Tomando orden del cliente 7
EL CLIENTE 7 YA NO TIENE MÁS QUE ORDENAR
..... Cliente 7 comiendo

```
- GUI Window (Simulación restaurante):**
 - Numero de Hilos (Clientes): 10
 - Numero de Meseros: 1
 - Numero de asientos: 7
 - Probabilidad de llegada (%): 80
 - Detener (button)
 - Comenzar (button)
 - Lista clientes sentados: 7 1 9 0 8
 - Lista clientes esperando:

Bottom Screenshot:

- Terminal Output:**

```

..... Cliente 3 comiendo
Llego el cliente % 7
.....EL cliente 3 se marcha
Mesero 4 atendiendo a 9
Tomando orden del cliente 9
EL CLIENTE 9 YA NO TIENE MÁS QUE ORDENAR
..... Cliente 9 comiendo
.....EL cliente 9 se marcha
Mesero 6 atendiendo a 6
Clientes sentados:6 -2 -4 -7 -
Clientes esperando: 1 -
Clientes sentados: 6 -2 -4 -7 -1 -
Clientes esperando:
Clientes sentados:6 -2 -4 -7 -1 -
Clientes esperando: 5 -
Clientes sentados: 6 -2 -4 -7 -1 -5 -
Clientes esperando:
Tomando orden del cliente 6
Llego el cliente % 1
EL CLIENTE 6 YA NO TIENE MÁS QUE ORDENAR
..... Cliente 6 comiendo
Llego el cliente % 5
.....EL cliente 6 se marcha
Mesero 5 atendiendo a 2
Clientes sentados:2 -4 -7 -1 -5 -
Clientes esperando: 3 -
Clientes sentados: 2 -4 -7 -1 -5 -3 -
Clientes esperando:
Tomando orden del cliente 2
Llego el cliente % 3

```
- GUI Window (Simulación restaurante):**
 - Numero de Hilos (Clientes): 10
 - Numero de Meseros: 1
 - Numero de asientos: 7
 - Probabilidad de llegada (%): 80
 - Detener (button)
 - Comenzar (button)
 - Lista clientes sentados: 2 4 7 1 5 3
 - Lista clientes esperando:

The screenshot shows a Kali Linux desktop environment. A terminal window is open, displaying the output of a Python script for a restaurant simulation. The script tracks the arrival, service, and departure of customers, as well as the status of waiters and tables. A graphical window titled 'Simulación restaurante' is also open, showing input fields for the number of threads (10), waiters (1), and tables (7), and a probability of arrival (80%). It includes 'Detener' and 'Comenzar' buttons, and displays the current state of the simulation, such as the list of seated customers (2 4 7 1 5 3) and the list of waiting customers (empty).

```
Archivo Editar Ver Buscar Terminal Ayuda
root@kali: /media/root/ENRIQUE

..... Cliente 3 comiendo
Llego el cliente % 7
.....EL cliente 3 se marcha
Mesero 4 atendiendo a 9
Tomando orden del cliente 9
EL CLIENTE 9 YA NO TIENE MÁS QUE ORDENAR
..... Cliente 9 comiendo
.....EL cliente 9 se marcha
Mesero 6 atendiendo a 6
Clientes sentados:6 -2 -4 -7 -
Clientes esperando: 1 -
Clientes sentados: 6 -2 -4 -7 -1 -
Clientes esperando:
Clientes sentados:6 -2 -4 -7 -1 -
Clientes esperando: 5 -
Clientes sentados: 6 -2 -4 -7 -1 -5 -
Clientes esperando:
Tomando orden del cliente 6
Llego el cliente % 1
EL CLIENTE 6 YA NO TIENE MÁS QUE ORDENAR
..... Cliente 6 comiendo
Llego el cliente % 5
.....EL cliente 6 se marcha
Mesero 5 atendiendo a 2
Clientes sentados:2 -4 -7 -1 -5 -
Clientes esperando: 3 -
Clientes sentados: 2 -4 -7 -1 -5 -3 -
Clientes esperando:
Tomando orden del cliente 2
Llego el cliente % 3
```

Simulación restaurante

Numero de Hilos (Clientes): 10

Numero de Meseros: 1

Numero de asientos: 7

Probabilidad de llegada (%): 80

Detener Comenzar

Lista clientes sentados 2 4 7 1 5 3

Lista clientes esperando