

SISTEMAS CONCURRENTE Y DISTRIBUIDOS

NOMBRE: José Alberto

APELLIDOS: Hoces Castro

DNI: ██████████

GRUPO: 1

JUSTIFICACIÓN: Para lograr que Yoshi no espere y siga produciendo tartas he hecho uso de MPI_Send. Este supone un envío asíncrono seguro y podemos borrar el buffer enviado para la siguiente tanda. Lo mismo con los equipos, pueden mandar el resultado de su tanda y seguir comiendo nuevas tandas producidas. Con MPI_Probe y MPI_Get_Count conseguimos el tamaño del buffer de tartas que se recibe.

CAPTURAS:

Inicio de ejecución y fin de ejecución

```
joshoc7@joshoc7-Aspire-A315-56:~/Escritorio/Tercero/scd-p3-fuentes$ mpicxx -std=c++11 -o sol_HocesCastroJoseAlberto_SCD_P3 sol_HocesCastroJoseAlberto_SCD_P3.cpp
joshoc7@joshoc7-Aspire-A315-56:~/Escritorio/Tercero/scd-p3-fuentes$ mpirun -oversubscribe -np 3 ./sol_HocesCastroJoseAlberto_SCD_P3
El equipo 2 se lleva 1 tartas en la tanda 1
Tarta 1 del equipo 2 de la tanda 2: 8
El equipo 1 se lleva 2 tartas en la tanda 1
El equipo 1 se lleva 2 tartas en la tanda 2
Tarta 1 del equipo 1 de la tanda 2: 5
Tarta 2 del equipo 1 de la tanda 2: 5
Equipo 2 se ha comido 2 kg en la tanda 1
El equipo 2 se lleva 1 tartas en la tanda 2
Tarta 1 del equipo 2 de la tanda 2: 4
Equipo 1 se ha comido 6 kg en la tanda 1
Tarta 1 del equipo 1 de la tanda 2: 9
Tarta 2 del equipo 1 de la tanda 2: 8
Equipo 2 se ha comido 1 kg en la tanda 2
Equipo 1 se ha comido 3 kg en la tanda 2
Gana el equipo Mario con 2 puntos
joshoc7@joshoc7-Aspire-A315-56:~/Escritorio/Tercero/scd-p3-fuentes$
```

El inicio y final de la ejecución se ven en la misma captura. Esto se debe a que he probado el programa para 2 rondas para poder ver a ojo si funcionaba bien. Si desea probarlo para más rondas, las rondas son la variable global x de mi programa.