



## **Proyecto de Simulación** *"Puerto Sobrecargado"*

**Est. Lázaro Raúl Iglesias Vera**

*C-412*

L.IGLESIAS@ESTUDIANTES.MATCOM.UH.CU

# Índice

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Modelación</b>	<b>3</b>
<b>3</b>	<b>Implementación</b>	<b>3</b>
3.1	Código . . . . .	3
3.2	Ideas generales . . . . .	3
3.3	Eventos . . . . .	4
3.4	Tiempos . . . . .	4
3.5	Ejecución . . . . .	4
<b>4</b>	<b>Datos obtenidos</b>	<b>4</b>

# 1

## Introducción

Conociendo particularidades de una zona portuaria, tales como la manera en que distribuyen los arribos al puerto, cantidad de muelles, cantidad de barcos a atender, así como otros datos presentes en la orientación [4] se desea conocer la media del tiempo de espera de los barcos en el puerto.

Dado el problema antes mencionado, será analizada una manera de simular el comportamiento del puerto de modo que se logre estimar dicha media.

# 2

## Modelación

Para simular el comportamiento descrito en la orientación se utilizó como base un modelo de varios servidores conectados en paralelo. Dichos servidores serían los muelles disponibles en el puerto, los cuales tendrán como usuarios a los barcos que llegan al puerto.

El flujo en el sistema queda bloqueado cada vez que se hace una operación de transportación de un barco, dado que solo se tiene un remolcador, es decir, si se analiza el sistema dividido en 3 secciones, entrada, servidores y salida, solo puede estar desplazándose un único barco entre secciones, lo cual no afecta el comportamiento propio de cada sección.

La entrada genera una cola de llegada de la cual serán desplazados ordenadamente los usuarios hacia los servidores, de donde, según van terminando de recibir su servicio, son trasladados a la salida.

El costo temporal de las operaciones en este sistema determina el tiempo de espera de cada barco y por tanto nos permitirá hallar la media de espera.

# 3

## Implementación

### 3.1 Código

La implementación del sistema se puede encontrar en [1] y será descrita en las siguientes secciones.

### 3.2 Ideas generales

Los eventos se manejarán ordenadamente, puesto que de no hacerse de esta manera podrían surgir problemas conocidos, como es el caso de **starvation** o **muerte por hambre**, pues algún barco puede quedar olvidado y esto afectar el resultado que se busca. Por lo anterior y la necesidad de conocer el tiempo de los eventos, se consideró crear un sistema donde los eventos son atendidos según su orden temporal.

Con este objetivo se debe mantener en el sistema una cola de eventos que los mantenga ordenados respecto al tiempo. No obstante, esto no garantiza que el sistema fluya de la manera esperada; puesto que cada evento está relacionado a un usuario, además de al tiempo en el que ocurre, lo cual no se está tomando en consideración. Si hay 2 eventos que ocurren al mismo tiempo, debería ser atendido primero aquel cuyo usuario haya arribado primero al sistema; sin embargo son atendidos según su aparición en la cola, lo que resulta incorrecto en algunos casos. Para solucionar este problema basta mantener una ordenación de los eventos que tenga en cuenta ambos criterios.

Para manejar los eventos se creó un controlador que mantiene la cola ordenada de la manera antes mencionada. Pero al atender un evento es necesario, en la mayoría de las ocasiones, generar uno nuevo que continúe el recorrido del usuario en cuestión dentro del sistema, lo cual no representa un problema si cada evento conoce cuál es el siguiente estado al que debe moverse el barco.

En conclusión, para simular el puerto basta definir un controlador que genere y ejecute los eventos correctamente apoyándose en la cola de las ocurrencias de los mismos.

Todo lo antes mencionado se puede encontrar en [3].

Otro detalle importante es que cada evento debe saber calcular su tiempo de ejecución, y para ello se utilizan las implementaciones brindadas en [5]

### 3.3 Eventos

Como ya quedó implícito, los eventos estarán compuestos por tres datos de interés y tendrán la siguiente estructura **<time, user, callback>**.

El recorrido de un barco en el puerto está determinado por los siguientes eventos:

*Arrive*  $\Leftrightarrow$  *Enqueue*  $\Rightarrow$  *Move*  $\Rightarrow$  *Dock*  $\Rightarrow$  *Ready*  $\Rightarrow$  *Depart*  $\Rightarrow$  *Done*

los cuales se encargan respectivamente de: generar un nuevo arribo, encolar el nuevo barco, moverlo hacia los muelles, recoger su cargamento, notificar que está listo para desocupar su muelle, salir nuevamente al puerto, y abandonarlo.

El significado de la transición *Arrive*  $\Leftarrow$  *Enqueue* no es más que el de añadir nuevos usuarios al sistema mientras esto sea posible. Esto se debe a que *Enqueue* es el evento que representa el momento en que llega un barco al puerto y *Arrive* es el encargado de generar dichos momentos. Por tanto, tras cada ocurrencia de *Enqueue*, se debe generar un nuevo arribo mientras no hayan llegado todos los usuarios que se planea atender. A pesar de la existencia de esta transición, se puede observar que el orden de ocurrencia de los eventos es lineal para cada barco.

### 3.4 Tiempos

Todas las distribuciones que describen el tiempo de duración de algún proceso fueron implementadas en [5] tal cual se muestra en el capítulo 5 de [2].

### 3.5 Ejecución

Para simular el puerto con 3 muelles, 1 remolcador y 10 barcos basta abrir una consola en la dirección del proyecto y ejecutar el comando **python harbor.py**

El sistema fue implementado de manera mas genérica, por lo cual tanto la cantidad de muelles como la de barcos puede ser modificada; incluso, puede modificarse la cantidad de veces que se va a simular para obtener un mayor volumen de datos antes de calcular la media.

Para ver como modificar dichos parámetros basta ejecutar **python harbor.py -help**

Ver README.md en [1] para más detalles.

## 4

### Datos obtenidos

Si se analizan las simulaciones del sistema, fijando en cada caso uno de los parámetros: barcos y muelles respectivamente, se puede observar que: el aumento o disminución en cuanto a la cantidad de barcos que arriban es directamente proporcional a la media; sin embargo, las variaciones en la cantidad de muelles influyen de manera inversa en la media de espera.

Este comportamiento es algo intuitivo pero no es del todo cierto, dado que mientras más muelles, aumenta la cantidad de viajes que da el remolcador sin estar transportando ningún barco, lo que añade un costo adicional al sistema, que puede llegar a ser determinante en la media de espera.

## Referencias

- [1] Repositorio del proyecto ([ir](#))
- [2] Ross 2010 A First Course in Probability 8th Ed ([abrir](#))
- [3] Implementación del puerto ([ir](#))
- [4] Descripción del Problema ([abrir](#))
- [5] Implementación de las distribuciones ([ir](#))