



# UNIVERSIDAD DE GRANADA

SIMULACIÓN DE SISTEMAS  
GRADO EN INGENIERÍA INFORMÁTICA

---

## PRÁCTICA 3

MODELOS DE SIMULACIÓN DINÁMICOS Y DISCRETOS

---

### **Autor**

José María Sánchez Guerrero

### **Rama**

Computación y Sistemas Inteligentes



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

CURSO 2019-2020

# Índice general

<b>1. Mi segundo modelo de simulación Discreto</b>	<b>2</b>
1.1. Simulación con incremento fijo de tiempo . . . . .	2
1.2. Simulación con incremento variable de tiempo . . . . .	3

## Capítulo 1

# Mi segundo modelo de simulación Discreto

Nuestro modelo de simulación consistirá en un servidor que presta un determinado servicio a una serie de clientes, los cuales solicitarán dicho servicio periódicamente. Cuando llega un cliente y el servidor no está ocupado, será atendido inmediatamente; en caso contrario, el cliente tendrá que esperar en la cola. Cuando se completa un servicio, el servidor elegirá al siguiente en una forma FIFO.

Al empezar la simulación, no habrá clientes esperando y el servidor está libre. Utilizaremos el mismo generador exponencial tanto para el tiempo que tardarán en llegar los clientes, como el tiempo que tardará el servidor en atender a cada uno.

### 1.1. Simulación con incremento fijo de tiempo

En esta simulación, vamos a tratar al tiempo incrementándolo de unidad en unidad. Para evitar problemas con el manejo del tiempo, tendremos que modificar los generadores de datos para que nos devuelvan los valores redondeados al entero más próximo. Si obtenemos un valor igual a 0, devolveremos 1 en su lugar, ya que el suceso generado quedaría en un tiempo anterior al actual, que generamos al incrementar en una unidad.

Este será nuestro código resultante, que nos servirá tanto para generar el tiempo de llegada del cliente como para generar el tiempo del servicio (sólo tendremos que modificar la variable *tlleg* por *tserv*):

```
1 float generallegada(float tllleg){
2     float u = random(); // o tambien rand() en lugar de random()
3     u = ( u / (RAND_MAX+1.0) ); //RAND_MAX es una constante del sistema
4     u = round( -tllleg * log(1-u) );
5
6     if (u != 0)
7         return u;
8     else
9         return 1.0;
10 }
```

Para la simulación vamos a emplear diferentes unida

## 1.2. Simulación con incremento variable de tiempo