**Universidad Politécnica de Cartagena**

**Escuela Técnica Superior de Ingeniería de Telecomunicación**

**PRÁCTICAS DE MODELADO Y SIMULACIÓN**

# BOLETÍN DE ENTREGA

# Práctica 4: Análisis estadístico de resultados

## INTEGRANTES DEL GRUPO:

|  |  |
| --- | --- |
| NOMBRE Y APELLIDOS | CORREO ELECTRÓNICO |
| Diego Ismael Antolinos García | diego.antolinos@edu.upct.es |
| Andrés Ruz Nieto | andres.ruz@edu.upct.es |

**Información y metodología de evaluación:**

## Esta práctica tiene un peso total del 10% en la evaluación de la asignatura

1. El boletín debe rellenarse y convertirse a **formato PDF** para su envío a través de aula virtual, en las fechas dispuestas en el calendario
2. La evaluación se basará en las **respuestas del boletín** y también se contempla la posibilidad de realizar **entrevistas online** individuales para verificar la autoría de las mismas.

## 

## 

1. **Proporcione las fórmulas/códigos solicitados en (1 pto/ejercicio):**
   1. **EJERCICIO 1**

Partimos de la ecuación que se nos da en el punto 6 de la práctica:

Despejamos la variable t de la ecuación de la tolerancia normalizada y sustituimos:

Posteriormente, despejamos en ambos lados:

Seguidamente, despejaríamos en ambos lados:

El término central se corresponde a una distribución Normal(0,1), al cual llamaremos Z:

Finalmente, sustituimos Z:

Por simetría podríamos poner lo siguiente:

* 1. **EJERCICIO 2**

function [cuasivarianza] = cuasivarianza(nummuestras, summuestras, sumcuadrado)

cuasivarianza = (1/(nummuestras-1))\*(sumcuadrado- (((summuestras/nummuestras)^2)\*nummuestras));

end

* 1. **EJERCICIO 3**

function [unomenosalfa, intizqda, intderecha] = calidad(tolrelativa, nummuestras, summuestras, sumcuadrado)

mediaaritmetica = summuestras/nummuestras;

S = cuasivarianza(nummuestras, summuestras, sumcuadrado);

tolabsoluta = tolrelativa \* (mediaaritmetica);

z = tolabsoluta/(sqrt(S/nummuestras));

unomenosalfa = 1 - (1-normcdf(z))\*2;

intizqda = mediaaritmetica - tolabsuluta;

intderecha = mediaaritmetica + tolabsoluta;

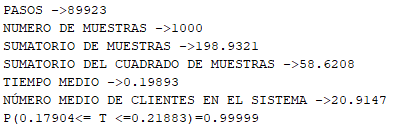
end

1. **Proporcione el código completo de su simulador, implementando todos los aspectos señalados en el enunciado de la práctica (4 puntos).**
2. **Indique, para cada una de las siguientes parametrizaciones: el intervalo de confianza para el tiempo medio de respuesta, el número de pasos, el número de muestras (**nummuestras**), el sumatorio de las muestras (**summuestras**)** **y el sumatorio del cuadrado de las muestras (**sumcuadrado**) obtenidos al completarse la simulación:**

**2.1. Configuración A (1.5 ptos)**

Z = 1

H = 10000

****D = 20

k = 5

tipoX=2

tipoS=2

param1X=100

param2X=0

param1S=21

param2S=0

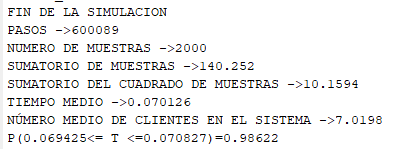
calidadobjetivo=0.9

tolrelativa=0.1

TEST=1000

**2.2. Configuración B (1.5 ptos)**

Z = 1

H = 100000

D = 50

k = 10

tipoX=2

tipoS=2

param1X=100

param2X=0

param1S=15

param2S=0

calidadobjetivo=0.95

tolrelativa=0.01

TEST=1000

%% ESQUELETO DE SIMULACION

listaEV = []; % Lista vacia al comienzo

t\_simulacion = 0.0; % Reloj de simulación

% ACCIONES DE INCIO: p.ej. definir estado, generar primeros eventos

% Se proporciona ejemplo del

% Caso cola de trabajos

t\_muestra = 1; %Seg de cada muestra

% TIPOS DE EVENTOS, CADA UNO UN NUMERO DIFERENTE

SALE = 0;

LLEGA = 1;

MUESTREO = 2;

% ESTADO

N = 0;

fifoTiempos = [];

% PARAMETROS DE SIMULACION

%1 -> Fishman-Moore

%2 -> Kobayashi

%3 -> Coveyou-McPherson

%4 -> glibc

%5 -> MMIX

generadorZ = 2;

Z = 1;

%PARAMETROS PARA M/M/5

calidadobjetivo = 0.9; % Suele estar entre el 90% y el 99%

tolrelativa = 0.1; % Suele estar entre el 1% y el 10%

TEST = 1000; % Cada cuantas muestras comprobamos calidad

H = 10000;

D = 20;

k = 5; % Numero de recursos

tipoX = 2;

param1X = 100;

param2X = 0;

tipoS = 2;

param1S = 21;

param2S = 0;

%PARAMETROS PARA M/M/10

% calidadobjetivo = 0.95; % Suele estar entre el 90% y el 99%

% tolrelativa = 0.01; % Suele estar entre el 1% y el 10%

% TEST = 1000; % Cada cuantas muestras comprobamos calidad

% H = 100000;

% D = 50;

% k = 10; % Numero de recursos

% tipoX = 2;

% param1X = 100;

% param2X = 0;

%

% tipoS = 2;

% param1S = 15;

% param2S = 0;

summuestrasT = 0;

muestrasT = 0;

summuestrasTcuadrado = 0;

nummuestrasT\_bloque = 0;

summuestrasT\_bloque = 0;

nt = 0;

summuestrasN = 0;

muestrasN = 0;

% PRIMEROS EVENTOS

[Z,taux] = aleatorio(Z,tipoX,param1X,param2X,generadorZ);

listaEV = encolarEvento(listaEV, taux, LLEGA,0);

[Z,taux] = aleatorio(Z,tipoX,param1X,param2X,generadorZ);

listaEV = encolarEvento(listaEV, taux, MUESTREO,0);

pasos = 0;

while true

pasos = pasos + 1;

[listaEV, tiempo, tipo, tllegadatarea] = sgteEvento(listaEV);

% Actualizamos el tiempo

t\_simulacion = tiempo;

switch tipo

case LLEGA

N = N+1;

[Z,taux] = aleatorio(Z,tipoX,param1X,param2X,generadorZ);

listaEV = encolarEvento(listaEV, t\_simulacion + taux, LLEGA,0);

if N<=k

[Z,taux] = aleatorio(Z,tipoS,param1S,param2S,generadorZ); % Tiempo en el recurso

listaEV = encolarEvento(listaEV, t\_simulacion + taux, SALE, t\_simulacion);

else

fifoTiempos = pushFIFO(fifoTiempos,t\_simulacion);

end

%DESCOMENTAR PARA EJECUCIÓN PASO A PASO

%display('LLEGADA');

%[t\_simulacion]

%pause

case SALE

N = N-1;

if N>=k % Otro trabajo pasa a ocupar el "procesador"

[fifoTiempos, tllegadacola] = popFIFO(fifoTiempos); % Recuperamos el primer tiempo en cola

[Z,taux] = aleatorio(Z,tipoS,param1S,param2S,generadorZ); % Tiempo en el recurso

listaEV = encolarEvento(listaEV, t\_simulacion + taux, SALE, tllegadacola);

end

%DESCOMENTAR PARA EJECUCIÓN PASO A PASO

%display('SALE');

%[t\_simulacion, tllegadatarea, t\_simulacion-tentrada]

%pause

nt = nt + 1;

if nt > H

summuestrasT\_bloque = summuestrasT\_bloque + (t\_simulacion - tllegadatarea);

nummuestrasT\_bloque = nummuestrasT\_bloque + 1;

if nummuestrasT\_bloque == D

summuestrasT = summuestrasT + summuestrasT\_bloque/nummuestrasT\_bloque;

muestrasT = muestrasT + 1;

summuestrasTcuadrado = summuestrasTcuadrado + (summuestrasT\_bloque/nummuestrasT\_bloque)^2;

nummuestrasT\_bloque = 0;

summuestrasT\_bloque = 0;

if ~mod(muestrasT,TEST)

[unomenosalfa, intizqda, intderecha] = calidad(tolrelativa, muestrasT, summuestrasT, summuestrasTcuadrado);

if(unomenosalfa >= calidadobjetivo)

break;

end

end

end

end

case MUESTREO

muestrasN=muestrasN+1;

summuestrasN=summuestrasN+N;

[Z,taux] = aleatorio(Z,tipoX,param1X,param2X,generadorZ);

listaEV = encolarEvento(listaEV, t\_simulacion + taux, MUESTREO,tllegadatarea);

end

end

% Mostramos los promedios calculador

disp('FIN DE LA SIMULACION');

disp(strcat('PASOS -> ',num2str(pasos)));

disp(strcat('NUMERO DE MUESTRAS -> ',num2str(muestrasT)));

disp(strcat('SUMATORIO DE MUESTRAS -> ',num2str(summuestrasT)));

disp(strcat('SUMATORIO DEL CUADRADO DE MUESTRAS -> ',num2str(summuestrasTcuadrado)));

disp(strcat('TIEMPO MEDIO -> ',num2str(summuestrasT / muestrasT)));

disp(strcat('NÚMERO MEDIO DE CLIENTES EN EL SISTEMA -> ',num2str(summuestrasN/muestrasN)));

disp(strcat('P(',num2str(intizqda),'<= T <=',num2str(intderecha),')=',num2str(unomenosalfa)));