

```

simulacion sistema de espera.txt
# AMPLIACIÓN DE INVESTIGACIÓN OPERATIVA
# I.T. EN INFORMÁTICA DE SISTEMAS Y GESTIÓN
# TEMA 4: SISTEMAS DE LÍNEAS DE ESPERA

# Programa para la simulación de un sistema de cola con modelo M|M|1

# Simulación asíncrona o orientada a objetos, es decir, observaremos
# el sistema cada vez que se produzca un suceso, esto es, o bien
# una llegada o una salida.

# Fijaremos un tiempo T máximo de funcionamiento del sistema.

# Función para el modelo M|M|1: simulacionMM1

# Argumentos:
# T: tiempo de observación del sistema
# ratiollegada: Tasa de llegada (parámetro de la exponencial)
# ratioservicio: Tasa de servicio (parámetro de la exponencial)

simulacionMM1<-function (T, ratiollegada, ratioservicio)
{
  # Simulación de la primera llegada.
  LL <- rexp(1, ratiollegada) # Generación del tiempo de llegada
  i <- 1 # Contador de clientes nuevos
  Tiempollegada <- LL # Vector de llegadas de clientes
  Tiemposervicio <- numeric() # Vector de servicio de clientes
  if (LL > T) {
    res <- c(0, 0, 0, 0)
  }
  else {
    t <- 0 # Iniciamos el tiempo del sistema
    n <- 0 # Iniciamos el número clientes en el sistema
    SS <- rexp(1, ratioservicio) # Generación del tiempo de servicio
    j <- 1 # Contador de clientes servidos
    lista <- list(llegada = LL, salida = SS + LL) # Tiempo de sucesos
    Tiempollegada <- c(LL) # Primera llegada
    Tiemposervicio <- c(SS) # Primer servicio
    res <- c(0, 0, 0, 0) # Iniciamos el proceso
  }
  # Simulación del sistema en general.
  while (t<T) { # Tiempo máximo de simulación
    # Determinación del próximo suceso
    tprox <- min(c(lista[[1]], lista[[2]])) # Tiempo
    suceso <- (1:2)[c(min(lista[[1]]), min(lista[[2]])) == tprox] # Tipo
    # Simulación de llegadas
    if (suceso == 1) { # Observación del suceso
      lista[[1]] <- lista[[1]][lista[[1]] != tprox]
      t <- tprox # Actualizamos el tiempo de sucesos
      n <- n + 1 # Un nuevo cliente en el sistema
      # Almacenamos el resultado en una matriz, donde
    }
  }
}

```

```

simulacion sistema de espera.txt
# cada fila indica la observación de un suceso,
# anotándose el tiempo de ocurrencia, tipo de
# suceso, tiempo de la llegada y número de
# clientes en el sistema.

res <- rbind(res, c(t, suceso, Tiempollegada[i],n))
# Próxima simulación
LL <- rexp(1, ratiollegada) # Generación de llegada
i <- i + 1 # Un nuevo individuo
Tiempollegada <- c(Tiempollegada, LL)
lista[[1]] <- c(lista[[1]], t + LL)
}
# Simulación de servicios
else { # Observación del suceso
  lista[[2]] <- lista[[2]][lista[[2]] != tprox]
  t <- tprox # Actualizamos el tiempo de sucesos
  n <- n - 1 # Un cliente abandona el sistema
  # Almacenamos el resultado en una matriz, donde
  # cada fila indica la observación de un suceso,
  # anotándose el tiempo de ocurrencia, tipo de
  # suceso, tiempo de la llegada y número de
  # clientes en el sistema.
  res <- rbind(res, c(t, suceso, Tiemposervicio[j], n))
  # Próxima simulación
  if (n > 0) { # Existen clientes en la cola
    SS <- rexp(1, ratioservicio) # Generación de servicio
    lista[[2]] <- c(lista[[2]], t + SS)
    j <- j + 1 # Nuevo servicio
    Tiemposervicio <- c(Tiemposervicio, SS) # Tiempo de servicio
  }
  else { # NO existen clientes en la cola
    SS <- rexp(1, ratioservicio)
    lista[[2]] <- c(lista[[2]], min(lista[[1]]) + SS)
    j <- j + 1
    Tiemposervicio <- c(Tiemposervicio, SS)
  }
}
}
}
}
res
}
# Simulación del ejemplo de las impresoras.
simulacionMM1(10,1,5)
# Salida función simulacionMM1:
# Una matriz. Cada fila es un suceso y las columnas como sigue:
# 1: Tiempo total del sistema
# 2: Tipo de suceso: 1 Llegada, 2 Salida
# 3: Tiempo de llegada o servicio

```

```

                                simulacion sistema de espera.txt
# 4: número de clientes en el sistema al producirse el evento.

# Análisis del sistema simulado.
simulacion<-simulacionMM1(1000,1,5)

# Número de llegadas al sistema
sum(simulacion[,2]==1)
# Número de salidas del sistema
sum(simulacion[,2]==2)
# Número de clientes en el sistema al terminar la simulación
sum(simulacion[,2]==1) - sum(simulacion[,2]==2)

# Estimación de los principales parámetros del modelo:

# Tasa de llegada
tasallegada<-function(simulacion){
1/mean(simulacion[simulacion[,2]==1,3])
}
tasallegada(simulacion);1

# Tasa de servicio
tasaservicio<-function(simulacion){
1/mean(simulacion[simulacion[,2]==2,3])
}
tasaservicio(simulacion);5

# Tasa de utilización del sistema
utilizacion<-function(simulacion){
n<-nrow(simulacion)
ocioso<-simulacion[,4]==0
ocioso1<-(1:(n-1))[ocioso[1:(n-1)]]+1
1-sum(simulacion[ocioso1,1]-simulacion[ocioso1-1,1])/simulacion[n,1]
}
utilizacion(simulacion); 0.20

# L: Número de clientes en el sistema
L<-function(simulacion){n<-length(simulacion[,1])

```

```

                                simulacion sistema de espera.txt
sum(simulacion[-n,4]*(simulacion[-1,1]-simulacion[-n,1])/simulacion[n,1]
}
L(simulacion);0.25

# Lq: Número de clientes en la cola
Lq<-function(simulacion){n<-length(simulacion[,1]);
cola<-simulacion[-n,4]>1 # No conocemos su tiempo en cola
puesto<-(1:(n-1))[cola]+1
sum((simulacion[cola,4]-1)*(simulacion[puesto,1]-simulacion[cola,1])/simulacion
[n,1]
}
Lq(simulacion);0.05

# W: Tiempo medio en el sistema
w<-function(simulacion){n<-length(simulacion[,1]);
llegada<-simulacion[,2]==1
salidas<-simulacion[,2]==2
nn<-min(sum(llegada),sum(salidas)) # clientes servidos
salidas1<-(1:n)[salidas]
llegadas1<-(1:n)[llegada]
sum(simulacion[salidas1[1:nn],1]-simulacion[llegadas1[1:nn],1])/nn
}
w(simulacion);0.25

# w: Tiempo en el sistema
w<-function(simulacion){n<-length(simulacion[,1]);
llegada<-simulacion[,2]==1
salidas<-simulacion[,2]==2
nn<-min(sum(llegada),sum(salidas)) # clientes servidos
salidas1<-(1:n)[salidas]
llegadas1<-(1:n)[llegada]
simulacion[salidas1[1:nn],1]-simulacion[llegadas1[1:nn],1]
}

```

```

simulacion sistema de espera.txt
hist(w(simulacion),prob=T)
lines(density(w(simulacion)))
ks.test(w(simulacion),"pexp",5-1) # distribución exponencial

# wq: Tiempo medio en la cola

wq<-function(simulacion){llegadacola<-(simulacion[,2]==1)&(simulacion[,4]>1)
salidascola<-(simulacion[,2]==2)&(simulacion[,4]>=1)
tllegada<-simulacion[llegadacola,1]
tsalidas<-simulacion[salidascola,1]

nn<-min(sum(llegadacola),sum(salidascola)) # clientes servidos que han esperado

sum(tsalidas[1:nn]-tllegada[1:nn])/sum(simulacion[,2]==2)
}

wq(simulacion);0.05

# wq: Tiempo en cola

wq<-function(simulacion){llegadacola<-(simulacion[,2]==1)&(simulacion[,4]>1)
salidascola<-(simulacion[,2]==2)&(simulacion[,4]>=1)
tllegada<-simulacion[llegadacola,1]
tsalidas<-simulacion[salidascola,1]

nn<-min(sum(llegadacola),sum(salidascola)) # clientes servidos que han esperado

tsalidas[1:nn]-tllegada[1:nn]
}

hist(wq(simulacion),prob=T)
lines(density(wq(simulacion)))
ks.test(wq(simulacion),"pexp",5-1) # distribución exponencial

# P0: Probabilidad de estar el sistema ocioso

p0<-function(simulacion){n<-length(simulacion[,1]);

```

```

simulacion sistema de espera.txt
cero<-simulacion[,4]==0
cero1<-(1:(n-1))[cero[-n]]+1
sum(simulacion[cero1,1]-simulacion[cero1-1,1])/simulacion[n,1]
}

p0(simulacion);0.80

# Comparar costo total para distintas tasas de servicio.

# ccs: Costo por cliente en el sistema.
# cus: costo por unidad de tasa de servicio.
# tsmin: tasa de servicio mínima.
# tsmax: tasa de servicio máxima.

tasaservicio<-function (tsmin, tsmax, densidad, ccs, cus, T, ratiollegada,
repeticiones)
{
ts <- seq(tsmin, tsmax, densidad)
res <- numeric()
log <- length(ts)
for (i in 1:log) {
res1 <- numeric()
for (j in 1:repeticiones) {
simulacion <- simulacionMM1(T, ratiollegada, ts[i])
res1 <- c(res1, L(simulacion) * ccs + cus * ts[i])
}
res <- c(res, mean(res1))
}
res
}

# Ejemplo:

tsmin<-3
tsmax<-6
densidad<-1
ccs<-10
cus<-5
tllegada<-2
repeticiones<-10
T<-1000

cts<-tasaservicio(tsmin, tsmax, densidad, ccs, cus, T, tllegada,
repeticiones)

plot(ts<-seq(tsmin,tsmax,densidad),cts,type="l",
ylab="Costo total esperado",xlab="Tasa de servicio",col=2) # Resultado
simulado

lines(ts,ts*cus+ccs*(tllegada/(ts-tllegada))) # Resultado teórico

```