

1.- SISTEMAS, MODELOS Y SIMULACIÓN

1.1- Algunos conceptos básicos

El proceso de interés se llama *sistema* y para estudiarlo se realizan una serie de hipótesis acerca de su funcionamiento, que constituyen el *modelo*. Si las relaciones matemáticas que componen el modelo son sencillas, pueden emplearse métodos matemáticos para obtener una solución analítica. En la mayoría de los casos, los modelos no pueden resolverse analíticamente y deben estudiarse mediante simulación, resolviendo numéricamente el modelo, con ayuda del ordenador.

Por ejemplo, consideremos el caso de un empresario que planea ampliar las instalaciones de su empresa, pero no está seguro de que la ganancia potencial de productividad justifique el coste de la aplicación. Ciertamente, no sería razonable que, para salir de dudas, realizara la ampliación y luego se volviera atrás si esta no resulta rentable. Una alternativa consiste en simular la operación de la empresa actual y la de la ampliada y comparar sus productividades.

Se define el *estado* de un sistema como la colección de variables necesarias para describir el sistema en un instante determinado, de acuerdo con los objetivos del estudio.

Por ejemplo, si desea estudiarse una oficina bancaria, para determinar el número de empleados necesarios para proporcionar un servicio adecuado a los clientes que desean realizar depósitos o reintegros, el sistema puede definirse como la porción del banco consistente en los empleados y los clientes haciendo cola para ser atendidos. Posibles variables de estado del modelo son el número de empleados ocupados, el número de clientes que hay en el banco y el instante de llegada de cada cliente al banco.

Como vemos, un modelo es la representación de un sistema desarrollada para el propósito específico del estudio. Los modelos de simulación (no los sistemas reales en sí) pueden clasificarse en estáticos o dinámicos, deterministas o estocásticos y discretos o continuos.

Un *modelo discreto* es aquel en el cual las variables de estado pueden cambiar de valor sólo un número finito de veces por unidad de tiempo. Un *modelo continuo* es aquel cuyas variables de estado pueden cambiar continuamente con el tiempo.

La decisión de realizar un modelo discreto o continuo depende del objetivo específico del estudio y no del sistema en sí. Por ejemplo, para modelar el flujo de tráfico, puede realizarse un modelo discreto si son relevantes las características y movimiento de los coches individuales. En caso contrario, puede realizarse un modelo continuo.

Un *modelo estático* representa el sistema en un determinado instante (por ejemplo, los modelos del tipo Monte Carlo). Un *modelo dinámico* representa la evolución del sistema en el tiempo.

Un modelo se dice *determinista* si no contiene variables aleatorias. En caso contrario, se dice *estocástico*. Los datos de salida de un modelo determinista son únicos para un conjunto de entradas dado, los de un modelo estocástico pueden ser, a su vez, aleatorios.

El aparente conflicto entre determinismo y aleatoriedad desaparece si asumimos que las teorías científicas no son descubrimientos de las leyes de la naturaleza sino invenciones de la mente humana. Sus consecuencias son presentadas en forma determinista si estamos interesados en examinar el resultado de realizar el experimento una única vez. En cambio, son presentadas en forma probabilística si lo estamos en examinar los resultados de repetir varias veces el experimento.

En lo sucesivo centraremos nuestra atención en los modelos discretos, dinámicos y estocásticos, a excepción de la breve mención que haremos de los métodos de Monte Carlo.

Uno de los objetivos más importantes del modelado es mejorar nuestra comprensión acerca de lo que realmente está sucediendo en el sistema. Si el modelo no refleja el conocimiento que se tiene sobre el sistema, probablemente el modelo será inútil.

Es preciso familiarizarse, desde un primer momento, con la idea de que algunos sistemas importantes no pueden ser modelados correctamente y que la simulación no resulta ni barata ni fácil de aplicar correctamente. Asimismo, no debe infraestimarse el coste de la recogida de datos ni del tiempo de computación requerido para producir resultados precisos.

La simulación no es un procedimiento de optimización. Proporciona respuestas del tipo “El coste de tomar la acción X es C”, pero no proporciona respuestas del tipo “el coste se minimiza si se toma la acción Y”. Pese a ello, es cierto que simulando repetidamente puede establecerse cual, de entre unas determinadas, es la mejor política y que existen métodos de sistematizar esta comparación entre alternativas, a fin de encontrar la óptima.

Supuesto que un modelo sea válido para nuestro propósito, cuanto más sencillo sea mejor. No debe complicarse innecesariamente el modelo. No debe realizarse un modelo muy detallado desde el primer momento, ya que será un modelo difícil y caro de desarrollar, depurar y ejecutar. Será difícil obtener los datos adecuados para poder evaluar todos los parámetros del modelo. La salida será difícil de explicar e interpretar.

Es una estrategia mejor aprender de la experiencia con modelos relativamente simples. Es conveniente programar modularmente, de modo que, si es preciso, pueda aumentarse fácilmente la complejidad de alguna parte del modelo.

El primer paso en un estudio de simulación es la definición de sus objetivos: las preguntas a responder, las políticas o diseños a evaluar, etc.

Una vez que el modelo ha sido programado y es ejecutado sin errores, deben interpretarse sus resultados. Si las salidas fluctúan estadísticamente, debe determinarse cuántas simulaciones son precisas para poder extraer conclusiones estadísticamente válidas.

Veamos a continuación tres ejemplos diferentes de modelado y simulación, que nos servirán para ilustrar las explicaciones subsiguientes.

1.2- Simulación de una cola atendida por un único empleado

Consideremos un servicio de atención al cliente compuesto por un único empleado. Si llega un cliente y el empleado esta ocupado, aquel se pone al final de la cola en espera de su turno. Si al llegar, el empleado esta desocupado, será atendido inmediatamente.

Cuando el empleado termina de atender a un cliente, éste se marcha y aquel comienza a atender al primero de la cola.

El objetivo de la simulación será estimar el tiempo medio que debe esperar un cliente en la cola, desde que llega hasta que es atendido, y el número medio de clientes que componen la cola.

El tiempo que transcurre entre la llegada de un cliente y la llegada del siguiente es una variable aleatoria de distribución de probabilidad F_A . Sea A_i el tiempo que transcurre entre la llegada del cliente (i-1)-ésimo y la del i-ésimo.

El tiempo de servicio al cliente i-ésimo, S_i , que es el tiempo que transcurre desde que comienza a ser atendido por el empleado hasta que acaba de ser atendido y se marcha, es una variable aleatoria de distribución de probabilidad F_S .

En general, las distribuciones de probabilidad F_A y F_S deben determinarse tomando datos del sistema y ajustándolos a una distribución. En este ejemplo supondremos que el periodo entre llegadas sucesivas de clientes son variables aleatorias IID (Independent Identically Distributed: "Idénticamente distribuidas" significa que obedecen a la misma distribución de probabilidad) exponenciales con una media de 1 minuto y que el tiempo de servicio a cada cliente son variables aleatorias IID exponenciales con una media de 0.5 minutos. Los tiempos de servicio y los intervalos entre llegadas sucesivas son independientes.

Las variables de estado del sistema pueden ser: el estado del empleado (libre u ocupado), el número de personas que esperan en la cola a ser atendidas, el instante de llegada de cada una de las personas que esperan en la cola y el instante en que se ha producido el último (más reciente) evento.

El estado del empleado es necesario para saber si el cliente puede ser atendido inmediatamente o por el contrario debe esperar haciendo cola. Obsérvese que si el número de clientes en la cola es distinto de cero el empleado estará forzosamente ocupado y si es cero puede estar libre u ocupado.

Debe conocerse el instante de llegada de cada cliente para poder calcular el tiempo que ha estado esperando en la cola, que es el periodo que transcurre desde que llega hasta que comienza a ser atendido. Si cuando llega el empleado esta desocupado, será atendido inmediatamente, con lo cual, su tiempo de espera será cero.

Los tiempos de espera de los clientes deben conocerse a fin de calcular, al finalizar la simulación, el tiempo medio de espera de un cliente en la cola. Para ello puede ir realizándose durante la simulación la suma de los tiempos de espera de los clientes. Al final de la simulación, dividiendo el tiempo total de espera por el número de clientes que han comenzado a ser atendidos, obtenemos el tiempo medio de espera.

Deben conocerse el número de integrantes de la cola en cada instante, así como el instante en que se ha producido el anterior evento, para ir calculando la integral respecto del tiempo del número de personas a la cola, a fin de obtener, al final de la simulación, el promedio de personas a la cola. Para ello, se dividirá la integral, calculada durante la simulación, del número de clientes a la cola en cada instante respecto del tiempo, por el tiempo simulado.

Hay dos tipos de eventos distintos en el sistema: la llegada de un nuevo cliente y la finalización de la atención a un cliente.

Si el empleado se encuentra libre, la llegada de un nuevo cliente, que es atendido inmediatamente, hace que pase a estar ocupado. Si, por el contrario, cuando llega el nuevo cliente el empleado esta ocupado, el número de integrantes de la cola aumentará en uno.

La marcha de un cliente supone que el empleado quedará libre si no hay nadie esperando en la cola. En caso contrario, el primero de la cola comenzará a ser atendido, reduciéndose en uno el número de integrantes de ésta.

En este ejemplo, los dos tipos de eventos modifican el estado del sistema, ya que cambian el estado del operario o el número de clientes que componen la cola. En general, esto no tiene porque ser así: un evento, por ejemplo, puede producir el fin de la simulación o forzar un cambio en el modo de operación del sistema que no suponga un cambio en su estado.

Debido a la naturaleza dinámica de los modelos de simulación de eventos discretos, debemos conocer el valor del tiempo simulado al avanzar la simulación y además necesitamos un mecanismo para hacer avanzar el tiempo simulado. Llamamos "reloj de la simulación" a la variable del modelo que va almacenando el tiempo simulado. Generalmente no existe relación entre el tiempo simulado y el tiempo necesario para ejecutar la simulación con el ordenador (salvo en sistemas en tiempo real).

Se usan normalmente dos métodos para hacer avanzar el reloj de la simulación:

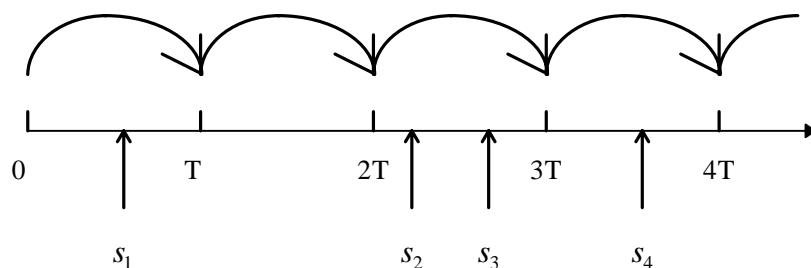
1.- *Avance hasta el instante del siguiente evento.* Se inicializa el reloj de la simulación a cero y se determinan los instantes en que se producirán los siguientes eventos. El reloj de la simulación es avanzado entonces hasta que ocurre el primero de ellos (el más inminente) y se actualiza el estado del sistema, de acuerdo con el evento, así como nuestro conocimiento acerca de los instantes de ocurrencia de los futuros eventos.

El reloj de simulación avanza hasta el siguiente evento, se actualiza el estado del sistema y se calculan los instantes de los siguientes eventos. Este proceso de avance del reloj de la simulación, de un evento al siguiente, se produce hasta que se alcanza alguna condición preestablecida de finalización.

Al saltar el reloj de un evento al siguiente se evitan los periodos de inactividad del sistema.

2.- *Avance del tiempo en incrementos fijos.* El reloj de la simulación avanza en incrementos fijos de longitud convenientemente escogida, T . Tras cada actualización del reloj, se comprueba si durante el pasado intervalo se ha producido algún evento. Se considera que todos los eventos ocurridos durante el intervalo han sucedido al final del mismo, actualizándose el estado del sistema en consecuencia.

El método se ilustra en la figura. Las flechas curvas representan el avance del reloj de la simulación y s_i ($i:1,2,\dots$) los instante en que sucede el evento i -ésimo.



En el intervalo de tiempo $[0,T)$ sucede un evento en el instante s_1 , pero el modelo considera que ha ocurrido en el instante T . En el intervalo $[T,2T)$ no suceden eventos, lo cual es comprobado por el modelo. En los instantes s_2 y s_3 , pertenecientes al intervalo $[2T,3T)$, suceden sendos eventos, considerándose que ambos han sucedido en el instante $3T$. Y así sucesivamente.

Debe establecerse un conjunto de reglas en el modelo a fin de poder decidir en qué orden se tratan los eventos cuando se considera que dos o más suceden en el mismo instante. Esto es una desventaja frente al método de avance hasta el siguiente evento, como también lo es la introducción del error debido al procesamiento de los eventos al final del intervalo en que suceden. Este último inconveniente pueden hacerse menos grave disminuyendo T , pero esto conlleva un aumento en el número de intervalos y por tanto en el número de comprobaciones de si se han producido o no eventos, lo cual supone un incremento en el tiempo de ejecución de la simulación.

Debido a estas consideraciones, el avance en incrementos de tiempo fijos no suele usarse cuando se presume que el tiempo entre sucesos puede variar considerablemente. Se usa preferiblemente cuando se sabe que los eventos suceden con una frecuencia fija conocida. Por ejemplo, si los datos en un sistema económico se obtienen con una frecuencia anual, parece natural hacer avanzar el reloj de la simulación a incrementos de tiempo, T , de 1 año.

Para la simulación de nuestro sistema emplearemos el método de avance al siguiente evento. Para ilustrarlo con un ejemplo establecemos la siguiente notación para las variables aleatorias:

t_i : instante de llegada del i -ésimo usuario ($t_0 = 0$)

$A_i = t_i - t_{i-1}$: intervalo de tiempo entre la llegada del usuario $(i-1)$ -ésimo y la del i -ésimo.

S_i : tiempo que tarda el empleado en atender al cliente i -ésimo, es decir, intervalo que transcurre desde que comienza hasta que acaba de atenderle, sin contar el tiempo que ha pasado en la cola.

D_i : tiempo de espera del cliente i -ésimo en la cola.

$C_i = t_i + D_i + S_i$: instante en que el cliente i -ésimo, tras ser atendido, se marcha.

s_i : instante de ocurrencia del evento i -ésimo (de cualquier tipo).

En el instante inicial de la simulación, $s_0 = 0$, el estado del empleado es desocupado y el instante en que se producirá la llegada del primer cliente se calcula generando un valor de la variable aleatoria A_1 , de la distribución F_A , y sumándolo a $s_0 = 0$, $t_1 = s_0 + A_1$.

El reloj de la simulación avanza entonces desde s_0 hasta el instante del primer evento, $s_1 = t_1$, que es la llegada del primer cliente.

Cada vez que llega un nuevo cliente se estima el tiempo que transcurrirá hasta la llegada del siguiente cliente. De la distribución F_A se obtiene A_2 , calculándose el instante en que llegará el segundo cliente $t_2 = t_1 + A_2$.

Como el primer cliente (que llega en el instante t_1) encuentra al empleado libre es atendido inmediatamente y el estado del empleado cambia de libre a ocupado.

Cada vez que el operario comienza a atender a un cliente se estima la duración que tendrá el servicio y, por tanto, el instante en que se producirá su marcha. De la distribución F_S se evalúa S_1 . Se calcula cuánto tiempo ha estado esperando en la cola, $D_1 = 0$. El instante en que el primer cliente se marchará será $C_1 = t_1 + D_1 + S_1$.

Así pues, en el instante t_1 , llegado el primer cliente, el empleado esta ocupado, la cola vacía y se conocen los instantes en que se producirá el siguiente evento de cada tipo: la llegada del segundo cliente, en t_2 , y la marcha del primer cliente, en C_1 .

Hay que decidir cual de estos dos eventos es más inminente, avanzando entonces el reloj de la simulación hasta él.

Si $t_2 < C_1$ el segundo cliente llegará antes de que el primero se haya marchado, con lo cual, el reloj de la simulación salta de $s_1 = t_1$ a $s_2 = t_2$.

El segundo cliente encontrará al empleado ocupado y deberá hacer cola. Se incrementa en 1 en número de clientes de la cola, pasando de 0 a 1. Se almacena el instante en que se pone a la cola para luego, cuando comience a ser atendido, poder saber el tiempo que ha pasado esperando en ella.

Cada vez que se produce la llegada de un nuevo cliente se estima el tiempo que transcurrirá hasta la llegada del siguiente. De F_A se obtiene A_3 , calculándose $t_3 = t_2 + A_3$.

Así pues, en t_2 , tras la llegada del segundo cliente, el empleado esta ocupado, hay un cliente en la cola y se conocen los instantes en que se producirá el siguiente evento de cada tipo: la marcha del primer cliente, en C_1 , y la llegada del tercer cliente, en t_3 . De nuevo debe decidirse cual de ellos es más inminente.

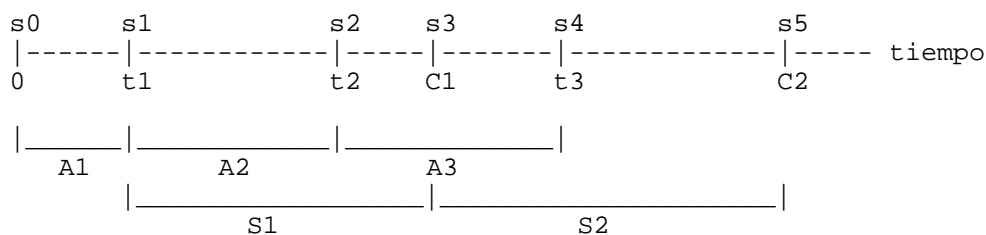
Si $C_1 < t_3$ el primer cliente se marchará antes de que llegue el tercer cliente. Así pues, el reloj de la simulación salta de $s_2 = t_2$ a $s_3 = C_1$.

La marcha del primer cliente deja al empleado desocupado, con lo cual el primer cliente de la cola (el segundo en llegar) comienza a ser atendido, volviendo a ocuparse el empleado.

Cada vez que un cliente abandona la cola para empezar a ser atendido se calcula el tiempo que ha consumido en ella, $D_2 = C_1 - t_2$.

Cada vez que un cliente comienza a ser atendido se estima, de la distribución F_S , el tiempo que durará su servicio, S_2 , calculándose el instante en que se marchará, $C_2 = C_1 + S_2$. El número de clientes de la cola disminuye en 1, pasando de 1 a 0.

Y así sucesivamente. Gráficamente:



La simulación terminará cuando se satisfaga cierta condición, por ejemplo, que el número de clientes que comienzan a ser atendidos supere cierto valor.

En nuestro caso, deseamos estimar el tiempo de espera medio de un cliente en la cola y el número medio de usuarios en la cola.

Para ello simulamos hasta obtener el tiempo de espera de n usuarios, D_1, D_2, \dots, D_n , así como el instante en el cual el cliente n -ésimo comienza a ser atendido, que será el tiempo necesario para que n usuarios completen su espera, $T_n = t_n + D_n$.

El tiempo de espera medio será:

$$\hat{d}(n) = \frac{\sum_{i=1}^n D_i}{n}$$

El promedio de usuarios en la cola:

$$\hat{Q}(n) = \frac{\int_0^{T_n} Q(t) dt}{T_n} = \frac{\sum_{i=1}^m R_i}{T_n}$$

donde:

$Q(t)$: número de clientes en la cola en el instante t

R_i : área del rectángulo bajo $Q(t)$ entre s_{i-1} y s_i si es el instante en que se produce el evento (de cualquier tipo) i -ésimo.

m : número de eventos que ocurren en el intervalo $[0, T_n]$

Así pues, el programa de simulación contendrá, entre otras, las siguientes variables:

- Estado del sistema: colección de variables de estado necesarias para describir el sistema en un instante determinado, de acuerdo con los objetivos del estudio. En nuestro caso, éstas pueden ser el estado (libre/ocupado) del empleado, el número de clientes de la cola, el instante de llegada de cada uno de ellos y el instante en que se produjo el anterior evento.
- Reloj de la simulación: variable que almacena el valor actual del tiempo simulado.
- Lista de eventos: contiene el instante en ocurrirá el evento más inminente de cada uno de los tipos posibles. En nuestro caso, contendrá el instante de llegada el siguiente cliente y el de marcha del que esta siendo atendido.
- Contadores estadísticos: son las variables usadas para almacenar la información estadística acerca del sistema, a partir de las cuales se obtendrán las variables cuya estimación es el objeto de la simulación. En nuestro caso éstas serán:
- El número total de clientes que ha comenzado a atender el empleado desde el comienzo de la simulación (todos los que ya se han marchado si esta libre y todos los que se han marchado más 1, el que esta actualmente atendiendo, si esta ocupado).

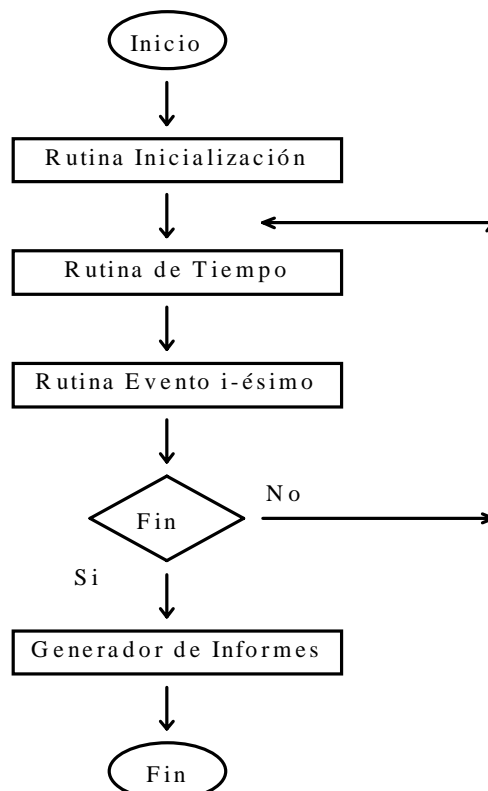
- La suma del tiempo de espera de todos los clientes que han comenzado a ser atendidos desde el comienzo de la simulación (todos los que ya se han marchado si esta libre y todos los que se han marchado más 1, el que esta actualmente atendiendo, si esta ocupado). Para calcular esta suma durante la simulación debe conocerse el tiempo de espera de cada cliente, motivo por el cual se guardan sus instantes de llegada.

- La integral temporal del número de clientes que hay en cada instante en la cola. Para ello, teniendo en cuenta que la función "número de clientes a la cola" es una función entera discontinua respecto al tiempo, cada vez que ocurre un evento se calcula la contribución a la integral del intervalo entre el evento anterior y éste, como el producto de la longitud del intervalo temporal por el número de personas que han estado en la cola durante él. Para ello se guardan el número de personas a la cola y el instante en que sucedió el evento anterior.

Asimismo, el programa estará constituido por las siguientes rutinas:

- Rutina de inicialización: empleada para inicializar el modelo en el instante cero.
- Rutina de tiempo: determina el siguiente evento (el más inminente) de los de la lista de eventos y avanza el reloj de la simulación hasta ese instante.
- Rutinas de eventos: son las rutinas, una para cada tipo de evento, que actualizan el estado del sistema cuando ocurre un evento y calculan el instante de ocurrencia del siguiente evento de cada tipo.
- Rutina generadora de informes: al finalizar la simulación calcula las medidas deseadas a partir de los contadores estadísticos e imprime un informe.
- Programa principal: distribuye el control entre las rutina anteriores. Al comienzo de la simulación llama a la rutina de inicialización. Realiza la llamada a la rutina de tiempo y a continuación a la rutina de eventos adecuada hasta que es detectado el final de la simulación. Al final de la simulación llama a la rutina de informes.

Un posible diagrama de flujo sería:



Veamos a continuación un programa, escrito en C, de simulación del sistema formado por 1 cola FIFO y 1 servidor. La razón de escoger un lenguaje de propósito general como es el C (es fundamental que el lenguaje permita gestionar dinámicamente la memoria) para realizar la simulación es de carácter pedagógica: creemos que si se aprende a simular en un lenguaje como C, en el cual debe prestarse atención hasta al más mínimo detalle, es menos probable que se cometan errores conceptuales cuando se empleen, más adelante, lenguajes de simulación de propósito general (SIMSCRIPT II.5 u otros) que son, en cierto modo, como "cajas negras" para el programador.

Para generar la variable aleatoria exponencial se obtiene primero una variable uniformemente distribuida sobre el intervalo $[0,1]$, $U(0,1)$, cuya función densidad de probabilidad vale uno para $x \in [0,1]$ y cero en cualquier otro caso.

La probabilidad de que una variable aleatoria $U(0,1)$ caiga en cualquier subintervalo $[x, x+\Delta x]$ contenido en $[0,1]$ es Δx . Por esto, a la variable aleatoria $U(0,1)$ se la llama uniforme. Cualquier variable aleatoria de interés puede obtenerse generando primero una o varias variables aleatorias uniformes $U(0,1)$.

En nuestro caso particular, generamos la variable aleatoria uniforme haciendo uso de la función

```
#include <stdlib.h>
int rand(void);
```

contenida en la librería `<stdlib.h>`, disponible en sistema UNIX y en ANSI C. Según se indica en el manual de referencia de la librería, `rand()` usa un generador de números aleatorios multiplicativo congruente con un periodo 2^{32} . Da como resultado un número pseudoaleatorio entero de 0 a `RAND_MAX`. La constante simbólica `RAND_MAX`, definida en `stdlib.h`, toma como valor $2^{15} - 1$, es decir, 32.767. Dividiendo `rand()` por 32767 obtenemos un número pseudoaleatorio comprendido en el intervalo $[0,1]$.

Para generar una variable aleatoria exponencial de media m , tomamos el logaritmo neperiano de la variable uniforme $U(0,1)$, v_unif , y multiplicamos el resultado por el negativo de m :

$$v_expon = -m \ln(v_unif)$$

Para entender el porqué, obsérvese que si X es una variable aleatoria exponencial con media b ($b>0$), entonces para cualquier $x \geq 0$ la probabilidad de que X sea menor o igual que x es:

$$P\{X \leq x\} = F(x) = \int_0^x f(t) dt = 1 - e^{-\frac{x}{b}}$$

donde $f(t)$ es la densidad de probabilidad de la variable aleatoria X :

$$f(t) = \frac{1}{b} e^{-\frac{t}{b}} \quad \text{para } x \geq 0, b > 0 \text{ media}$$

y $F(x)$ es la distribución de probabilidad acumulada de X .

Si U es una variable aleatoria uniforme $U(0,1)$, entonces para cualquier $x \geq 0$:

$$P\{-b \ln(U) \leq x\} = P\left\{U \geq e^{-\frac{x}{b}}\right\} = 1 - e^{-\frac{x}{b}} = F(x)$$

de donde la variable aleatoria $X = -b \ln(U)$ tiene una distribución exponencial de media b .

Las rutinas de manipulación de colas están almacenadas en el fichero cola.h, el cual es incluido en nuestro programa mediante un #include.

Consta de las siguientes funciones:

<i>Esta_vacia:</i>	Informa acerca de si la cola posee algún elemento.
<i>Anadir_elemento:</i>	Añade un nuevo elemento al final de la cola.
<i>Sacar_elemento:</i>	Extrae el primer elemento de la cola.
<i>Imprime_cola:</i>	Realiza un listado del contenido de la cola.

Fichero cola.h:

```
#include <stdio.h>
#include <stdlib.h>
#include <alloc.h>

enum booleana {false,true};
typedef enum booleana booleana;

struct cola {
    float          dato;
    struct cola    *siguiente;
};

struct gestion_cola {
    struct cola    *inicio,
                  *fin;
};

booleana Esta_vacia(g_cola)
struct gestion_cola *g_cola;
{
    if (g_cola->inicio==NULL)
        return(true);
    else
        return(false);
}

booleana Anadir_elemento(g_cola,dato)
struct gestion_cola *g_cola;
float          dato;
{
    booleana      error;
    if (Esta_vacia(g_cola)==true)
        g_cola->inicio = g_cola->fin =
            (struct cola *) calloc(1,sizeof(struct cola));
    else
        g_cola->fin= (g_cola->fin)->siguiente =
            (struct cola *) calloc(1,sizeof(struct cola));
    if (g_cola->fin==NULL) {
        printf("\nERROR: No hay memoria disponible\n");
        error=true;
    }
    else {
        (g_cola->fin)->dato=dato;
        error=false;
    };
    return(error);
}

booleana Sacar_elemento(g_cola,dato)
struct gestion_cola *g_cola;
float          *dato;
{
    struct cola    *aux;
    booleana      error;
```

```

if (Esta_vacia(gCola)==false) {
    *dato=(gCola->inicio)->dato;
    aux=gCola->inicio;
    if ((gCola->inicio)->siguiente==NULL)
        gCola->inicio=gCola->fin=NULL;
    else
        gCola->inicio=(gCola->inicio)->siguiente;
    free(aux);
    error=false;
}
else {
    printf("\nERROR: intento de sacar un elemento de una cola vacia\n");
    error=true;
};
return(error);
}

void Imprime_cola(gCola)
struct gestion_cola *gCola;
{
    struct cola *aux;
    if (Esta_vacia(gCola)==true) {
        printf("Cola vacia");
    }
    else {
        aux=gCola->inicio;
        do {
            printf("%f\t",aux->dato);
            aux=aux->siguiente;
        } while (aux!=NULL);
    };
    printf("\n");
    return;
}

```

El programa contiene las siguientes funciones:

- Rutina generadora de números aleatorios:
Distr_expon: genera una variable aleatoria exponencial.
- Inicialización:
Inicialización: inicialización del reloj de la simulación, el estado del sistema, los contadores estadísticos, la lista de eventos y del generador de números aleatorios.
- Rutina de tiempo:
Rutina_tiempo: avanza el reloj de la simulación hasta el evento más inminente de los de la lista de eventos.
- Rutinas de eventos:
Evento_llegada: actualiza el estado del sistema, la lista de eventos y los contadores estadísticos.
Evento_marcha: actualiza el estado del sistema, la lista de eventos y los contadores estadísticos.
- Rutina de decisión sobre la finalización de la simulación:
Final_simulación: devuelve true si se satisface alguna de las condiciones de finalización de la simulación y false en caso contrario.
- Rutinas I/O:
Entrada_datos: entrada por teclado de los valores medios de las distribuciones exponenciales (intervalo entre llegadas y tiempo de servicio) y del número de clientes que comienzan a ser atendidos que desea simularse.
Informe: a partir de los contadores estadísticos calcula las variables de interés (tiempo medio de espera en la cola y número medio de clientes en la cola) y lista sus valores.
Imprime_estado_simulación: realiza un listado del valor actual del reloj de la simulación, de las variables de estado del sistema, de los contadores estadísticos y la de lista de eventos.
- Programa principal:

main: llama a las rutinas de inicialización y de entrada de datos. Mientras la rutina de finalización no indique que la simulación ha concluido llama sucesivamente a la rutina de tiempo y de eventos correspondiente. Al final de la simulación llama a la rutina de informe.

Listado del programa:

```
#include <time.h>
#include <math.h>
#include <conio.h>
#include <dos.h>
#include "a:\cola.h"

#define NO_DEFINIDO -1.0

enum status_servidor {libre,ocupado};
typedef enum status_servidor status_servidor;

enum tipo_evento {llegada,marcha};
typedef enum tipo_evento tipo_evento;

struct estado_sistema {
    status_servidor      estado_employed;
    unsigned int         n_clientes_cola;
    struct gestion_cola  g_cola;
    float                t_evento_anterior;
};

struct lista_eventos {
    float                t_prox_llegada,
                        t_marcha;
};

struct contad_estadisticos {
    unsigned int         n_clientes_inic_atencion;
    float                t_total_espera,
                        integ_n_clientes_cola;
};

struct media_var_aleat {
    float                interv_llegadas,
                        t_servicio;
};

float Distr_expon(media)
float      media;
{
    float      v_unif,
              v_expon;
    do {
        v_unif=rand()/32727.0;
    } while (!(v_unif>0));
    v_expon=-media*log(v_unif);
    return(v_expon);
}

void Inicializacion (reloj,estado,eventos,cont_est,media)
float      *reloj;
struct media_var_aleat      media;
struct estado_sistema      *estado;
struct lista_eventos        *eventos;
struct contad_estadisticos  *cont_est;
{
    /* Inicializacion del reloj de la simulacion */
    *reloj=0;
    /* Inicializacion del estado del sistema */
    estado->estado_employed=libre;
    estado->n_clientes_cola=0;
    estado->t_evento_anterior=0;
    (estado->g_cola).inicio=NULL;
    (estado->g_cola).fin=NULL;
    /* Inicializacion de la lista de eventos */
    eventos->t_prox_llegada=*reloj+Distr_expon(media.interv_llegadas);
    eventos->t_marcha=NO_DEFINIDO;
    /* Inicializacion de los contadores estadisticos */
    cont_est->n_clientes_inic_atencion=0;
}
```

```

cont_est->t_total_espera=0;
cont_est->integ_n_clientes cola=0;
/* Inicializacion del generador de numeros aleatorios */
randomize();
return;
}

tipo_evento Rutina_tiempo (reloj,eventos)
float          *reloj;
struct lista_eventos  eventos;
{
if (eventos.t_prox_llegada < eventos.t_marcha ||
    eventos.t_marcha==NO_DEFINIDO) {
    *reloj=eventos.t_prox_llegada;
    return(llegada);
}
else {
    *reloj=eventos.t_marcha;
    return(marcha);
}
};

booleana Evento_llegada (reloj,estado,eventos,cont_est,media)
float          reloj;
struct estado_sistema      *estado;
struct lista_eventos      *eventos;
struct contad_estadisticos *cont_est;
struct media_var_aleat     media;
{
booleana          error=false;
eventos->t_prox_llegada=reloj+Distr_expon(media.interv_llegadas);
if (estado->estado_empleado==libre) {
    estado->estado_empleado=ocupado;
    cont_est->n_clientes_inic_atencion++;
    eventos->t_marcha=reloj+Distr_expon(media.t_servicio);
}
else {
    cont_est->integ_n_clientes cola+=(estado->n_clientes cola)*
        (reloj-estado->t_evento_anterior);
    error=Anadir_elemento(&(estado->g cola),reloj);
    (estado->n_clientes cola)++;
};
estado->t_evento_anterior=reloj;
return(error);
}

booleana Evento_marcha (reloj,estado,eventos,cont_est,media)
float          reloj;
struct media_var_aleat     media;
struct estado_sistema      *estado;
struct lista_eventos      *eventos;
struct contad_estadisticos *cont_est;
{
float          *t_llego;
booleana          error=false;
if (estado->n_clientes cola==0) {
    estado->estado_empleado=libre;
    eventos->t_marcha=NO_DEFINIDO;
}
else {
    eventos->t_marcha=reloj+Distr_expon(media.t_servicio);
    error=Sacar_elemento(&(estado->g cola),t_llego);
    cont_est->t_total_espera+=reloj-(*t_llego);
    cont_est->integ_n_clientes cola+=(estado->n_clientes cola)*
        (reloj-estado->t_evento_anterior);
    (estado->n_clientes cola)--;
    cont_est->n_clientes_inic_atencion++;
};
estado->t_evento_anterior=reloj;
return(error);
}

booleana Final_simulacion(n_clientes_inic_atencion,n_clientes_fin)
unsigned int  n_clientes_inic_atencion,

```

```

        n_clientes_fin;
    {
        if (n_clientes_inic_atencion < n_clientes_fin)
            return(false);
        else
            return(true);
    }

void Entrada_datos (media_va,n_clientes_fin)
struct media_var_aleat      *media_va;
unsigned int                *n_clientes_fin;
{
    clrscr();
    printf("\tEntrada de datos:\n");
    printf("\t-----\n");
    printf("\nMedia del intervalo entre llegadas: ");
    scanf("%f",&(media_va->interv_llegadas));
    printf("\nMedia del tiempo de servicio: ");
    scanf("%f",&(media_va->t_servicio));
    printf("\nNum. clientes cuyo tiempo de espera debe simularse (1-65535): ");
    scanf("%d",n_clientes_fin);
    return;
}

void Informe (reloj,cont_est)
float                reloj;
struct contad_estadisticos    cont_est;
{
    float                t_espera_medio,
                        nCola_medio;
    printf("\nFin de la simulacion...\n\n");
    printf("\tI N F O R M E\n");
    printf("\t-----\n\n");
    t_espera_medio=cont_est.t_total_espera/cont_est.n_clientes_inic_atencion;
    printf("\nTiempo medio de espera en la cola : %f\n\n",t_espera_medio);
    nCola_medio=cont_est.integ_n_clientesCola/reloj;
    printf("Numero medio de clientes en la cola: %f\n\n",nCola_medio);
    return;
}

void Imprime_estado_simulacion(reloj,estado,eventos,cont_est)
float                reloj;
struct estado_sistema        estado;
struct lista_eventos        eventos;
struct contad_estadisticos    cont_est;
{
    printf("-----\t\tReloj: %f\n",reloj);
    if (estado.estado_empleado==libre)
        printf("Empleado LIBRE ");
    else
        printf("Empleado OCUPADO");
    printf("\t\t%d clientes en la cola",estado.n_clientesCola);
    printf("\t\tUltimo evento: %f\nCola:\t",estado.t_evento_anterior);
    ImprimeCola(&(estado.gCola));
    printf("Lista de eventos: Llegada = %f  Marcha = ",
            eventos.t_prox_llegada);
    if (eventos.t_marcha==NO_DEFINIDO)
        printf("- \n");
    else
        printf("%f\n",eventos.t_marcha);
    printf("Cont. estadisticos:\n");
    printf("\t\tNumero de clientes que han empezado a ser atendidos:\t%d\n",
            cont_est.n_clientes_inic_atencion);
    printf("\t\tTiempo total de espera de los clientes en la cola:\t%f\n",
            cont_est.t_total_espera);
    printf("\t\tIntegral num. clientes en la cola respecto del tiempo:\t%f\n",
            cont_est.integ_n_clientesCola);
}

main()
{
    struct media_var_aleat      media_va;
    struct estado_sistema        estado;
    struct lista_eventos        eventos;

```

```

struct contad_estadisticos      cont_est;
unsigned int                    n_clientes_fin;
float                           reloj;
booleana                       error;
Entrada_datos(&media_va,&n_clientes_fin);
Inicializacion(&reloj,&estado,&eventos,&cont_est,media_va);
Imprime_estado_simulacion(reloj,estado,eventos,cont_est);
do {
if (Rutina_tiempo(&reloj,eventos)==llegada)
    error=Evento_llegada(reloj,&estado,&eventos,&cont_est,media_va);
else
    error=Evento_marcha(reloj,&estado,&eventos,&cont_est,media_va);
Imprime_estado_simulacion(reloj,estado,eventos,cont_est);
}
while (Final_simulacion(cont_est.n_clientes_inic_atencion,
    n_clientes_fin)==false && error==false);
if (error==false)
    Informe(reloj,cont_est);
else
    printf("Terminacion anormal...\n");
return;
}

```

Veamos un ejemplo de la evolución del sistema durante la simulación y el resultado obtenido (el tiempo puede considerarse, en el ejemplo, medido en minutos):

```

Entrada de datos:
-----

Media del intervalo entre llegadas:      1.0
Media del tiempo de servicio:           0.5
Num. clientes cuyo tiempo de espera debe simularse (1-65535): 5

-----
Reloj: 0.000000
Empleado LIBRE      0 clientes en la cola Ultimo evento: 0.000000
Cola: Cola vacia
Lista de eventos: Llegada = 4.549548  Marcha = -
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      0
    Tiempo total de espera de los clientes en la cola: 0.000000
    Integral num. clientes en la cola respecto del tiempo: 0.000000

-----
Reloj: 4.549548
Empleado OCUPADO    0 clientes en la cola Ultimo evento: 4.549548
Cola: Cola vacia
Lista de eventos: Llegada = 6.371961  Marcha = 4.796673
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      1
    Tiempo total de espera de los clientes en la cola: 0.000000
    Integral num. clientes en la cola respecto del tiempo: 0.000000

-----
Reloj: 4.796673
Empleado LIBRE      0 clientes en la cola Ultimo evento: 4.796673
Cola: Cola vacia
Lista de eventos: Llegada = 6.371961  Marcha = -
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      1
    Tiempo total de espera de los clientes en la cola: 0.000000

```

```

Integral num. clientes en la cola respecto del tiempo:      0.000000

-----
Reloj: 6.371961
Empleado OCUPADO      0 clientes en la cola Ultimo evento: 6.371961
Cola: Cola vacia
Lista de eventos: Llegada = 6.494414  Marcha = 6.453289
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      2
    Tiempo total de espera de los clientes en la cola: 0.000000
    Integral num. clientes en la cola respecto del tiempo:      0.000000

-----
Reloj: 6.453289
Empleado LIBRE      0 clientes en la cola Ultimo evento: 6.453289
Cola: Cola vacia
Lista de eventos: Llegada = 6.494414  Marcha = -
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      2
    Tiempo total de espera de los clientes en la cola: 0.000000
    Integral num. clientes en la cola respecto del tiempo:      0.000000

-----
Reloj: 6.494414
Empleado OCUPADO      0 clientes en la cola Ultimo evento: 6.494414
Cola: Cola vacia
Lista de eventos: Llegada = 7.640840  Marcha = 7.127680
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      3
    Tiempo total de espera de los clientes en la cola: 0.000000
    Integral num. clientes en la cola respecto del tiempo:      0.000000

-----
Reloj: 7.127680
Empleado LIBRE      0 clientes en la cola Ultimo evento: 7.127680
Cola: Cola vacia
Lista de eventos: Llegada = 7.640840  Marcha = -
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      3
    Tiempo total de espera de los clientes en la cola: 0.000000
    Integral num. clientes en la cola respecto del tiempo:      0.000000

-----
Reloj: 7.640840
Empleado OCUPADO      0 clientes en la cola Ultimo evento: 7.640840
Cola: Cola vacia
Lista de eventos: Llegada = 8.381031  Marcha = 8.910286
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      4
    Tiempo total de espera de los clientes en la cola: 0.000000
    Integral num. clientes en la cola respecto del tiempo:      0.000000

-----
Reloj: 8.381031
Empleado OCUPADO      1 clientes en la cola Ultimo evento: 8.381031
Cola: 8.381031
Lista de eventos: Llegada = 8.597755  Marcha = 8.910286
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      4
    Tiempo total de espera de los clientes en la cola: 0.000000
    Integral num. clientes en la cola respecto del tiempo:      0.000000

-----
Reloj: 8.597755
Empleado OCUPADO      2 clientes en la cola Ultimo evento: 8.597755
Cola: 8.381031      8.597755
Lista de eventos: Llegada = 12.388444  Marcha = 8.910286
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      4
    Tiempo total de espera de los clientes en la cola: 0.000000
    Integral num. clientes en la cola respecto del tiempo:      0.216724

-----
Reloj: 8.910286
Empleado OCUPADO      1 clientes en la cola Ultimo evento: 8.910286
Cola: 8.597755
Lista de eventos: Llegada = 12.388444  Marcha = 9.028479
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      5
    Tiempo total de espera de los clientes en la cola: 0.529255
    Integral num. clientes en la cola respecto del tiempo:      0.841785

```

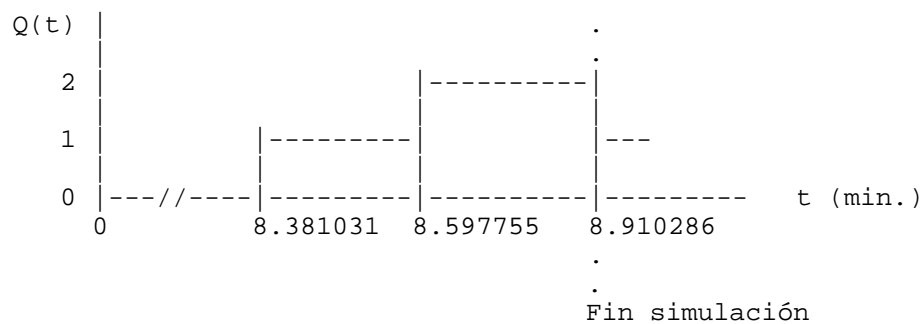
Fin de la simulacion...

I N F O R M E -----

Tiempo medio de espera en la cola : 0.105851
Numero medio de clientes en la cola: 0.094473

En esta simulación un único cliente abandona la cola para ser atendido: se pone a la cola en el instante 8.381031 min. y comienza a ser atendido en el instante 8.910286 min., que coincide con el final de la simulación. Al final de la simulación había un cliente en la cola, si bien, como no han comenzado a atenderle antes del final de la simulación, no influye en el valor de los contadores estadísticos.

Podemos representar $Q(t)$ respecto al tiempo, t :



Por tanto, el tiempo medio de espera en la cola se calcula dividiendo el tiempo que ha estado esperando el único cliente que ha sido atendido después de hacer cola, $(8.910286 - 8.381031) = 0.529255$ min., por el número de clientes que han comenzado a ser atendidos:

$$(0.529255 \text{ min}) / (5 \text{ clientes}) = 0.105851 \text{ min/cliente}$$

El número medio de clientes en la cola se calcula integrando en el tiempo el número de clientes a la cola y dividiendo por el tiempo simulado. Entre los instantes 0 y 8.381031 min. la cola ha estado vacía, con lo cual ese intervalo no contribuye a la integral. Entre 8.381031 min. y 8.597755 min. había un cliente en la cola, con lo cual, la contribución a la integral es:

$$(1 \text{ cliente}) * (8.597755 \text{ min.} - 8.381031 \text{ min.}) = 0.216724 \text{ cliente*min.}$$

Entre los instantes 8.597755 min. y 8.910286 min. había 2 clientes en la cola, con lo cual la contribución a la integral de este intervalo es:

$$(2 \text{ clientes}) * (8.910286 \text{ min.} - 8.597755 \text{ min.}) = 0.625062 \text{ cliente*min.}$$

El valor total de la integral es:

$$0.216724 + 0.625062 = 0.841786 \text{ cliente*min.}$$

Dividiendo por el tiempo de simulación, 8.910286 min., obtenemos el número medio de clientes en la cola:

$$(0.841786 \text{ cliente*min.}) / (8.910286 \text{ min.}) = 0.094473 \text{ clientes}$$

Repetiendo la simulación se obtiene un resultado diferente:

```

Entrada de datos:
-----

Media del intervalo entre llegadas:      1.0
Media del tiempo de servicio:           0.5
Num. clientes cuyo tiempo de espera debe simularse (1-65535):  5

-----
Reloj: 0.000000
Empleado LIBRE      0 clientes en la cola Ultimo evento: 0.000000
Cola: Cola vacia
Lista de eventos:  Llegada = 4.549548  Marcha = -
Cont. estadísticos:
    Numero de clientes que han empezado a ser atendidos:      0
    Tiempo total de espera de los clientes en la cola:  0.000000
    Integral num. clientes en la cola respecto del tiempo:    0.000000

-----
Reloj: 4.549548
Empleado OCUPADO    0 clientes en la cola Ultimo evento: 4.549548
Cola: Cola vacia
Lista de eventos:  Llegada = 4.731661  Marcha = 4.584194
Cont. estadísticos:
    Numero de clientes que han empezado a ser atendidos:      1
    Tiempo total de espera de los clientes en la cola:  0.000000
    Integral num. clientes en la cola respecto del tiempo:    0.000000

-----
Reloj: 4.584194
Empleado LIBRE      0 clientes en la cola Ultimo evento: 4.584194
Cola: Cola vacia
Lista de eventos:  Llegada = 4.731661  Marcha = -
Cont. estadísticos:
    Numero de clientes que han empezado a ser atendidos:      1
    Tiempo total de espera de los clientes en la cola:  0.000000
    Integral num. clientes en la cola respecto del tiempo:    0.000000

-----
Reloj: 4.731661
Empleado OCUPADO    0 clientes en la cola Ultimo evento: 4.731661
Cola: Cola vacia
Lista de eventos:  Llegada = 4.785454  Marcha = 6.048357
Cont. estadísticos:
    Numero de clientes que han empezado a ser atendidos:      2
    Tiempo total de espera de los clientes en la cola:  0.000000
    Integral num. clientes en la cola respecto del tiempo:    0.000000

-----
Reloj: 4.785454
Empleado OCUPADO    1 clientes en la cola Ultimo evento: 4.785454
Cola: 4.785454
Lista de eventos:  Llegada = 5.448385  Marcha = 6.048357
Cont. estadísticos:
    Numero de clientes que han empezado a ser atendidos:      2
    Tiempo total de espera de los clientes en la cola:  0.000000
    Integral num. clientes en la cola respecto del tiempo:    0.000000

-----
Reloj: 5.448385
Empleado OCUPADO    2 clientes en la cola Ultimo evento: 5.448385
Cola: 4.785454      5.448385
Lista de eventos:  Llegada = 5.829074  Marcha = 6.048357
Cont. estadísticos:
    Numero de clientes que han empezado a ser atendidos:      2
    Tiempo total de espera de los clientes en la cola:  0.000000
    Integral num. clientes en la cola respecto del tiempo:    0.662931

-----
Reloj: 5.829074
Empleado OCUPADO    3 clientes en la cola Ultimo evento: 5.829074
Cola: 4.785454      5.448385      5.829074
Lista de eventos:  Llegada = 5.994754  Marcha = 6.048357
Cont. estadísticos:
    Numero de clientes que han empezado a ser atendidos:      2
    Tiempo total de espera de los clientes en la cola:  0.000000

```

```

Integral num. clientes en la cola respecto del tiempo:      1.424310

-----
Reloj: 5.994754
Empleado OCUPADO      4 clientes en la cola Ultimo evento: 5.994754
Cola:  4.785454      5.448385      5.829074      5.994754
Lista de eventos: Llegada = 6.400387  Marcha = 6.048357
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      2
    Tiempo total de espera de los clientes en la cola:  0.000000
    Integral num. clientes en la cola respecto del tiempo:      1.921350
-----
Reloj: 6.048357
Empleado OCUPADO      3 clientes en la cola Ultimo evento: 6.048357
Cola:  5.448385      5.829074      5.994754
Lista de eventos: Llegada = 6.400387  Marcha = 7.292892
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      3
    Tiempo total de espera de los clientes en la cola:  1.262903
    Integral num. clientes en la cola respecto del tiempo:      2.135758
-----
Reloj: 6.400387
Empleado OCUPADO      4 clientes en la cola Ultimo evento: 6.400387
Cola:  5.448385      5.829074      5.994754      6.400387
Lista de eventos: Llegada = 8.212457  Marcha = 7.292892
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      3
    Tiempo total de espera de los clientes en la cola:  1.262903
    Integral num. clientes en la cola respecto del tiempo:      3.191851
-----
Reloj: 7.292892
Empleado OCUPADO      3 clientes en la cola Ultimo evento: 7.292892
Cola:  5.829074      5.994754      6.400387
Lista de eventos: Llegada = 8.212457  Marcha = 7.492419
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      4
    Tiempo total de espera de los clientes en la cola:  3.107409
    Integral num. clientes en la cola respecto del tiempo:      6.761869
-----
Reloj: 7.492419
Empleado OCUPADO      2 clientes en la cola Ultimo evento: 7.492419
Cola:  5.994754      6.400387
Lista de eventos: Llegada = 8.212457  Marcha = 7.887635
Cont. estadisticos:
    Numero de clientes que han empezado a ser atendidos:      5
    Tiempo total de espera de los clientes en la cola:  4.770754
    Integral num. clientes en la cola respecto del tiempo:      7.360450

```

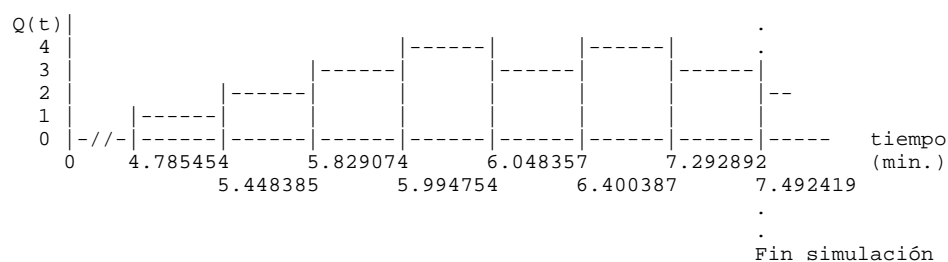
Fin de la simulacion...

I N F O R M E

Tiempo medio de espera en la cola : 0.954151

Numero medio de clientes en la cola: 0.982386

Podemos representar la función número de clientes en la cola, $Q(t)$, frente al tiempo:



El tiempo medio de espera en la cola se ha calculado teniendo en cuenta que los 3 clientes que abandonan la cola para ser atendidos tienen:

se pone a la cola	comienza atención	tiempo espera
4.785454 min.	6.048357 min.	1.262903 min.
5.448385 min.	7.292892 min.	1.844507 min.
5.829074 min.	7.492419 min.	1.663345 min.

	tiempo total de espera:	4.770755 min.

Dividiendo el tiempo total de espera en la cola por el número de clientes que han comenzado a ser atendidos se obtiene el tiempo medio de espera en la cola por cliente:

$$(4.770755 \text{ min}) / (5 \text{ clientes}) = 0.954151 \text{ min/cliente}$$

El valor de la integral de $Q(t)$ se calcula:

intervalo de tiempo (minutos)	longitud intervalo	num. clientes en la cola	contribución a la integral
0.000000 - 4.785454	4.785454	0	0.000000
4.785454 - 5.448385	0.662931	1	0.662931
5.448385 - 5.829074	0.380689	2	0.761378
5.829074 - 5.994754	0.165680	3	0.497040
5.994754 - 6.048357	0.053603	4	0.214412
6.048357 - 6.400387	0.352030	3	1.056090
6.400387 - 7.292892	0.892505	4	3.570020
7.292892 - 7.492419	0.199527	3	0.598581

		Valor de la integral:	7.360352

El número medio de clientes en la cola se calcula dividiendo el valor de la integral por el tiempo de simulación: $(7.360352 \text{ clientes} \cdot \text{min}) / (7.492419 \text{ min}) = 0.982373 \text{ clientes}$

En muchos estudios de simulación se desean obtener las características del modelo en estado estacionario, esto es, las características del modelo cuando la simulación se ha estado ejecutando durante un periodo de tiempo muy largo (teóricamente infinito).

En este caso, es posible calcular analíticamente el tiempo medio de espera en la cola y el número medio de clientes en la cola, obteniéndose, para los valores de las medias de las distribuciones tomados en el ejemplo, que ambas valen 0.5 minutos (Gross, D. and C. M. Harris: Fundamentals of Queuing Theory, Wiley, New York 1974, página 58).

Como vemos, los resultados obtenidos se alejan mucho de los verdaderos en el estacionario, debido a que hemos simulado para un número demasiado pequeño de clientes. Asimismo, los resultados obtenidos al repetir varias veces la simulación difieren considerablemente entre si.

Si repetimos 10 veces la simulación imponiendo como condición de finalización que el número de clientes que comienzan a ser atendidos sea 1000, para los mismos datos de entrada y calculamos el valor medio de los valores obtenidos, obtenemos:

Entrada de datos:

Media del intervalo entre llegadas: 1.0
Media del tiempo de servicio: 0.5
Num. clientes cuyo tiempo de espera debe simularse (1-65535): 1000

Resultados:

número simulación	1	2	3	4	5	6	7	8	9	10	Media
tiempo medio espera	0.570	0.490	0.441	0.467	0.559	0.523	0.335	0.493	0.588	0.585	0.505
número medio clientes en cola	0.604	0.475	0.416	0.466	0.570	0.527	0.326	0.477	0.601	0.627	0.509

Como vemos, si el objetivo de nuestra simulación fuera estimar estas medidas en el estacionario, nuestra elección de la condición de finalización (1000 clientes) nos habría permitido obtener unos valores bastante próximos a los verdaderos. Sin embargo, no debemos olvidar que la condición de finalización la hemos impuesto arbitrariamente. En la práctica la elección de qué condición de finalización nos dará una buena estimación de las características en el estacionario es muy difícil.

Para ilustrar esto supongamos que repetimos la simulación para una frecuencia de llegada de usuarios un 80% mayor (de 1 por minuto a 1.8 por minuto).

Entrada de datos:

```
-----
Media del intervalo entre llegadas:      0.556 minutos
Media del tiempo de servicio:           0.500 minutos
Num. de clientes cuyo tiempo de espera desea simularse (1-65535): 1000
```

El resultado verdadero, calculado analíticamente es:

```
Tiempo medio de espera en la cola:      4.5 minutos
Número medio de clientes en la cola:    8.1 minutos
```

Simulando 10 veces obtenemos los siguientes resultados:

número simulación	1	2	3	4	5	6	7	8	9	10	Media
tiempo medio de espera	5.692	2.664	3.739	2.530	4.349	4.626	2.844	4.059	3.666	6.954	4.112
número medio de clientes en cola	10.434	4.886	6.624	4.640	7.403	8.824	5.175	7.521	6.582	12.831	7.492

que son sensiblemente diferentes de los verdaderos. Parece claro que la condición de finalización no puede escogerse arbitrariamente.

En nuestro ejemplo la simulación finaliza cuando el número de clientes que han comenzado a ser atendidos supera una valor preestablecido, con lo cual el valor del reloj de la simulación al final de ésta es una variable aleatoria. Sin embargo, en muchas simulaciones es preferible detener su ejecución cuando el reloj de la simulación alcanza o supera un determinado valor. Por ejemplo, 480 minutos, que son la 8 horas que dura la jornada laboral del empleado. Después de las 8 horas no se permite la llegada de nuevos clientes, prosiguiendo la simulación hasta que todos los clientes que se encontraban a la cola han sido atendidos. En este caso, el número total de clientes que han comenzado a ser atendidos será una variable aleatoria.

1.3- Elección de la política de gestión del inventario de un único producto.

Consideremos una compañía que vende un único producto. Sus clientes ordenan pedidos a intervalos de tiempo aleatorios y de tamaño aleatorio. Un estudio experimental acerca de las actividades de la compañía aconseja:

- Considerar que los intervalos de tiempo entre pedidos sucesivos son variables aleatorias IID exponenciales de media 0.1 mes.
- Considerar que los tamaños de los pedidos que hacen a la compañía sus clientes, D , son variables aleatorias IID, independientes de los instantes en que se producen, de la forma:

$D=1$	tiene probabilidad $1/6$
$D=2$	tiene probabilidad $2/6$
$D=3$	tiene probabilidad $2/6$
$D=4$	tiene probabilidad $1/6$

A comienzo de cada mes, la compañía, a la vista de su inventario de existencias, debe decidir qué cantidad de producto compra a su suministrador. Para ello debe tener en cuenta que:

- costes de pedido ($K+i*Z$):
 - el precio de cada unidad de producto son $i=300$ pts.
 - cada vez que se hace un pedido, con independencia de las unidades de producto de que conste, deben pagarse $K=3200$ pts en concepto de gastos de tramitación.
- tiempo que tardará en recibirse el pedido:

experimentalmente se ha encontrado que el tiempo que transcurre desde que se ordena un pedido al suministrador hasta que se recibe el producto, es decir, el tiempo de servicio del suministrador, puede considerarse una variable aleatoria uniformemente distribuida en el intervalo de 0.5 mes a 1 mes.

La compañía sigue el siguiente criterio para decidir cuánto producto, Z , encargar a su suministrador:

$$Z = \begin{cases} S - I & \text{si } I < s \\ 0 & \text{si } I \geq s \end{cases}$$

donde:

Z : número de unidades de producto encargadas.

I : número de unidades en inventario al comienzo del mes, antes de hacer el pedido.

s, S : parámetros cuyo valor depende de la política de gestión del inventario de la empresa. El objetivo de la simulación es encontrar, de entre un conjunto de pares (s, S) posibles, el más rentable. Para ello se simula el inventario, durante un periodo de n meses, para comparar el coste medio mensual de cada una de las diferentes políticas de gestión.

Cuando un cliente solicita a la compañía una cantidad de producto que no supera lo inventariado, la demanda es atendida en el momento.

Si, por el contrario, la demanda supera las existencias de la compañía, se sirve inmediatamente la totalidad de lo inventariado, satisfaciéndose el exceso de demanda cuando la llegada de nuevo producto, encargado al suministrador, lo permita. En este caso, la nueva cantidad inventariada (la antigua menos la cantidad demandada), será un entero negativo. Cuando la compañía recibe la mercancía encargada a su suministrador, se satisfacen las demandas pendientes y el producto restante pasa al inventario.

Aparte del coste del pedido, la compañía debe afrontar los costes de mantenimiento del producto en inventario (almacenaje, seguros, impuestos, conservación, etc.) y los costes de déficit de inventario, es decir, del hecho de no poder servir a un cliente su pedido (fundamentalmente cuantifica la pérdida de prestigio frente al cliente que esto supone).

Para definir estos costes debemos introducir la siguiente notación:

$I(t)$: número de productos inventariados en el instante t . Es un entero positivo, negativo o cero. Los instantes de tiempo en que $I(t)$ decrece son aquellos en los que el cliente realiza la demanda.

$I^+(t) = \max\{I(t), 0\}$: número de productos que se encuentran realmente en inventario en el instante t .

$I^-(t) = \max\{-I(t), 0\}$: demanda no satisfecha en el instante t , es decir, producto pedido por el cliente y que no ha podido ser servido por falta de inventario.

Supondremos que el coste promedio de mantenimiento para los n meses es:

$$h \frac{\int_0^n I^+(t) dt}{n}$$

donde:

$h = 100$ pts por unidad de producto y por mes.

$\frac{\int_0^n I^+(t) dt}{n}$ es el promedio temporal del número de las unidades de producto actualmente en inventario.

Supondremos que el coste promedio del déficit de inventario para los n meses es:

$$p \frac{\int_0^n I^-(t) dt}{n}$$

donde:

$p = 500$ pts por unidad de producto y mes.

$\frac{\int_0^n I^-(t) dt}{n}$ es el promedio temporal de unidades de producto demandadas por los clientes y que no han podido ser servidas por falta de inventario.

Como condición inicial para la simulación, suponemos que hay 60 unidades de producto en inventario, $I(0)=60$, y que no se ha realizado ningún pedido.

Simulamos el inventario del sistema durante $n=120$ meses (10 años) y empleamos el coste total medio mensual obtenido (que es la suma del coste medio de pedido por mes, el coste medio de mantenimiento por mes y el coste medio de la demanda insatisfecha por mes) para comparar las nueve siguientes políticas de pedidos.

s	20	20	20	20	40	40	40	60	60
S	40	60	80	100	60	80	100	80	100

Las variables de estado del modelo de simulación de este sistema de inventario pueden ser :

- la cantidad de producto en inventario, $I(t)$
- la cantidad de producto encargado por la compañía a su suministrador y que todavía no ha sido servido.
- el instante en que se produjo el último cambio en el inventario (necesario para calcular las áreas bajo las funciones $I^+(t)$ y $I^-(t)$).

Consideramos 4 tipos de eventos:

- la llegada a la compañía de producto encargado al suministrador. Al producirse el evento debe actualizarse el cálculo del área bajo las funciones $I^+(t)$ y $I^-(t)$, incrementar el inventario con el producto recibido y dejar indefinido, en la lista de eventos, el instante en que se producirá la próxima recepción de producto procedente del suministrador.
- un cliente realiza un pedido a la compañía. Esto implica un cambio en el inventario, con lo cual debe actualizarse el cálculo del área bajo las funciones $I^+(t)$ y $I^-(t)$. Debe darse un valor a la variable aleatoria "cuantía del pedido" y restar el producto pedido del existente en inventario. Esto puede dar lugar, si la cantidad demandada supera la existente en inventario, a que el nuevo valor del inventario sea negativo. Debe calcularse el instante en que se producirá el siguiente pedido, sumando al reloj de la simulación el valor que tome la variable aleatoria exponencial "intervalo entre pedidos de clientes"
- el fin de la simulación, tras un tiempo simulado de n meses. Debe actualizarse el cálculo del área bajo las funciones $I^+(t)$ y $I^-(t)$.
- al comienzo de cada mes se realiza la evaluación del inventario y, si es necesario (de acuerdo con la política de gestión, definida por (s,S)), el pedido de producto a la compañía suministradora. En el caso en que $I < s$, se realiza un pedido al suministrador de $S-I$ unidades de producto, con lo cual, debe actualizarse el valor de los costes de pedido y determinar el instante en que se recibirá el producto (sumando al reloj de la simulación el valor que tome la variable uniformemente distribuida en el intervalo 0.5 - 1 mes "tiempo de servicio del suministrador"). Se realice ($I < s$) o no ($I \geq s$) un pedido, el próximo evento de este tipo se producirá al comienzo del siguiente mes.

Cuando se produzcan en el mismo instante de tiempo los eventos "fin de la simulación" y "evaluación de inventario", se dará prioridad a aquél. En general, el modelo debe diseñarse de modo que se establezca, entre los eventos que pueden suceder simultáneamente, el orden en que deben procesarse en caso de que esto ocurra.

Pueden definirse los siguientes contadores estadísticos:

- coste total de pedido. Necesario para calcular el coste medio del pedido mensual.
- área bajo la función $I^+(t)$. Necesario para calcular el coste medio mensual de mantenimiento del excedente.
- área bajo la función $I^-(t)$. Necesario para calcular el coste medio mensual de la demanda insatisfecha.

El objetivo de la simulación será la estimación del valor de las siguientes variables:

- coste medio total mensual. Suma de los costes medios mensuales de los pedidos, del mantenimiento del inventario y de la demanda insatisfecha.
- coste medio mensual de los pedidos.
- coste medio mensual de mantenimiento del inventario.
- coste medio mensual de la demanda insatisfecha.

El modelo consta de los siguientes parámetros:

- coeficientes s y S de la política de inventario. Deben introducirse en cada simulación.
- coeficientes h y p , multiplicadores que aparecen en las expresiones del coste promedio del mantenimiento del inventario y de la demanda insatisfecha respectivamente. Se considera $h=100$ y $p=500$.
- precios del suministrador: precio de cada unidad de producto (i) y precio de la tramitación del pedido (K). Se considera $i=300$ y $K=3200$.
- duración de la simulación en meses, n . Debe introducirse en cada simulación.
- media de la variable aleatoria exponencial "intervalo entre las sucesivas demandas de producto de los clientes a la empresa". Se considera igual a 0.1 meses.
- probabilidad de que un cliente realice un pedido de m unidades de producto, $D(m)$. Se considera $D(1)=1/6$, $D(2)=2/6$, $D(3)=2/6$, $D(4)=1/6$.
- cantidad de producto en inventario al comenzar la simulación, $I(0)$. Se considera $I(0)=60$.

El modelo de simulación contiene tres variables aleatorias de diferente tipo:

- Intervalo de tiempo entre dos pedidos consecutivos de producto de un cliente a la compañía. Es una variable exponencial, v_exp , de media $media_exp=0.1$ meses. Se genera a partir de una variable uniforme $U(0,1)$ de la forma:

$$v_exp = -media_exp \ln(U)$$

- Número de unidades de producto, D , de que consta el pedido realizado por el cliente a la compañía:

$D=1$	con probabilidad $1/6$
$D=2$	con probabilidad $2/6$
$D=3$	con probabilidad $2/6$
$D=4$	con probabilidad $1/6$

Se genera a partir de una variable uniforme $U(0,1)$. Si al evaluar la variable uniforme ésta toma un valor entre 0 y $1/6$, se dará a D el valor 1. Si toma un valor entre $1/6$ y $3/6$, se dará a D el valor 2, y así sucesivamente. Esto es:

$U(0,1)$ en el intervalo	Valor asignado a D
$[0,1/6)$	1
$[1/6,3/6)$	2
$[3/6,5/6)$	3
$[5/6,1]$	4

- El tiempo de servicio del suministrador, es decir, el tiempo que transcurre desde que la compañía ordena el pedido hasta que lo recibe, es una variable uniformemente distribuida en el intervalo $[0.5,1]$ mes, $U(0.5,1)$. Se genera a partir de la variable uniforme $U(0,1)$ de la forma:

$$U(0.5,1) = 0.5 + 0.5 * U(0,1)$$

En general, una variable uniforme en el intervalo $[A,B]$, $U(A,B)$, puede generarse a partir de una variable uniforme en $[0,1]$, $U(0,1)$, de la forma:

$$U(A,B) = A + (B - A) * U(0,1)$$

A continuación se lista un programa en C que realiza la simulación de este sistema.

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
#include <dos.h>
#include <math.h>

#define NO_DEFINIDO -1.0

enum tipo_evento {
    llegada_prod_suministrador,
    cliente_realiza_pedido,
    fin_simulacion,
    evaluacion_inventario
};

typedef enum tipo_evento tipo_evento;

struct estado_sistema {
    int          inventario,
              prod_pedido_suministrador;
    float        t_ultimo_cambio_inventario;
};

struct lista_eventos {
    float  t_llegada_producto,
          t_cliente_pedido,
          t_fin_simulacion,
          t_evaluacion_inventario;
};

struct contad_estadisticos {
    float  coste_total_pedido,
          integ_prod_inventario,
          integ_demanda_no_satisfecha;
};

struct v_a_discreta {
    float  valor_variable,
          probabilidad_valor_variable;
};

struct politica_inventario {
    int    S,
          s;
};

struct precio_suministrador {
    float  gestion,
          unidad_producto;
};

struct costes {
    struct precio_suministrador  pedido;
    float                        mantenimiento,
                                deficit;
};

struct intervalo {
    float  inicio,
          fin;
};

float Distr_expon(media)
float  media;
{
    float  v_unif,
          v_expon;
    do {
        v_unif=rand()/32727.0;
    } while (!(v_unif>0));
    v_expon=-media*log(v_unif);
    return(v_expon);
}

```

```

float Uniforme_interv_A_B(interv)
struct intervalo    interv;
/* Generacion U(a,b) */
{
float    v_unif;
v_unif=rand()/32727.0;
return(interv.inicio+(interv.fin-interv.inicio)*v_unif);
}

float Var_aleat_discreta(var_aleat_discreta)
struct v_a_discreta    *var_aleat_discreta;
{
float
    v_unif,
    prob_acumulada=0;
struct v_a_discreta    *aux;
v_unif=rand()/32727.0;
if (v_unif>1.0) v_unif=1.0;
aux=var_aleat_discreta;
while (v_unif>(prob_acumulada+aux->probabilidad_valor_variable)) {
    prob_acumulada+=aux->probabilidad_valor_variable;
    aux++;
}
return(aux->valor_variable);
}

void Inicializacion (reloj,estado,eventos,cont_est,t_fin_simulacion,
    media_interv_pedidos_cliente,interv_unif,
    unidades_pedidas,coste)
float
    *reloj,
    t_fin_simulacion,
    *media_interv_pedidos_cliente;
struct estado_sistema
struct lista_eventos
struct contad_estadisticos
struct v_a_discreta
struct intervalo
struct costes
    *estado;
    *eventos;
    *cont_est;
    unidades_pedidas[];
    *interv_unif;
    *coste;
{
/* Inicializacion del reloj de la simulacion */
*reloj=0;
/* Variable aleatoria "intervalo entre pedidos sucesivos de clientes" */
*media_interv_pedidos_cliente=0.1;
/* Variable aleatoria "cantidad de producto pedido por el cliente" */
unidades_pedidas[0].valor_variable=1.0;
unidades_pedidas[0].probabilidad_valor_variable=0.16667;
unidades_pedidas[1].valor_variable=2.0;
unidades_pedidas[1].probabilidad_valor_variable=0.33333;
unidades_pedidas[2].valor_variable=3.0;
unidades_pedidas[2].probabilidad_valor_variable=0.33333;
unidades_pedidas[3].valor_variable=4.0;
unidades_pedidas[3].probabilidad_valor_variable=0.16667;
/* Variable aleatoria "tiempo de servicio del suministrador" */
interv_unif->inicio=0.5;
interv_unif->fin=1.0;
/* Costes */
coste->pedido.gestion=3200.0;
coste->pedido.unidad_producto=300.0;
coste->mantenimiento=100.0;
coste->deficit=500.0;
/* Inicializacion del estado del sistema */
estado->inventario=60;
estado->prod_pedido_suministrador=0;
estado->t_ultimo_cambio_inventario=0;
/* Inicializacion de la lista de eventos */
eventos->t_llegada_producto=NO_DEFINIDO;
eventos->t_cliente_pedido=*reloj+Distr_expon(*media_interv_pedidos_cliente);
eventos->t_fin_simulacion=t_fin_simulacion;
eventos->t_evaluacion_inventario=0;
/* Inicializacion de los contadores estadisticos */
cont_est->coste_total_pedido=0;
cont_est->integ_prod_inventario=0;
cont_est->integ_demanda_no_satisfecha=0;
/* Inicializacion del generador de numeros aleatorios */
randomize();
return;
}

```

```

tipo_evento Rutina_tiempo (reloj,eventos)
float          *reloj;
struct lista_eventos  eventos;
{
if ( ( eventos.t_fin_simulacion <= eventos.t_llegada_producto ||
      eventos.t_llegada_producto == NO_DEFINIDO )      &&
      eventos.t_fin_simulacion <= eventos.t_cliente_pedido      &&
      eventos.t_fin_simulacion <= eventos.t_evaluacion_inventario) {
    *reloj=eventos.t_fin_simulacion;
    return(fin_simulacion);
}
else {
    if ( eventos.t_llegada_producto != NO_DEFINIDO      &&
        eventos.t_llegada_producto < eventos.t_fin_simulacion &&
        eventos.t_llegada_producto <= eventos.t_cliente_pedido &&
        eventos.t_llegada_producto <= eventos.t_evaluacion_inventario) {
        *reloj=eventos.t_llegada_producto;
        return(llegada_prod_suministrador);
    }
    else {
        if (eventos.t_cliente_pedido < eventos.t_fin_simulacion      &&
            (eventos.t_cliente_pedido < eventos.t_llegada_producto ||
             eventos.t_llegada_producto == NO_DEFINIDO)      &&
            eventos.t_cliente_pedido<=eventos.t_evaluacion_inventario
            ) {
            *reloj=eventos.t_cliente_pedido;
            return(cliente_realiza_pedido);
        }
        else {
            *reloj=eventos.t_evaluacion_inventario;
            return(evaluacion_inventario);
        };
    };
};
}

void Actualiza_integrales(reloj,estado,cont_est)
float          reloj;
struct estado_sistema      *estado;
struct contad_estadisticos *cont_est;
{
if (estado->inventario>0)
    cont_est->integ_prod_inventario+=estado->inventario*
        (reloj-estado->t_ultimo_cambio_inventario);
if (estado->inventario<0)
    cont_est->integ_demanda_no_satisfecha+=-estado->inventario*
        (reloj-estado->t_ultimo_cambio_inventario);
estado->t_ultimo_cambio_inventario=reloj;
return;
}

void Evento_fin_simulacion (reloj,estado,cont_est)
float          reloj;
struct estado_sistema      *estado;
struct contad_estadisticos *cont_est;
{
Actualiza_integrales(reloj,estado,cont_est);
return;
}

void Evento_llegada_producto (reloj,estado,eventos,cont_est)
float          reloj;
struct estado_sistema      *estado;
struct lista_eventos      *eventos;
struct contad_estadisticos *cont_est;
{
Actualiza_integrales(reloj,estado,cont_est);
estado->inventario+=estado->prod_pedido_suministrador;
estado->prod_pedido_suministrador=0;
eventos->t_llegada_producto=NO_DEFINIDO;
return;
}

```

```

void Evento_pedido_cliente (reloj,estado,eventos,cont_est,
    distr_num_prod_demandados,media_intervalo_pedidos_cliente)
float
    reloj,
    media_intervalo_pedidos_cliente;
struct estado_sistema
    *estado;
struct lista_eventos
    *eventos;
struct contad_estadisticos
    *cont_est;
struct v_a_discreta
    *distr_num_prod_demandados;
{
    Actualiza_integrales(reloj,estado,cont_est);
    estado->inventario -= (int) Var_aleat_discreta(distr_num_prod_demandados);
    eventos->t_cliente_pedido=reloj+
        Distr_expon(media_intervalo_pedidos_cliente);
    return;
}

void Evento_evaluacion_inventario(reloj,estado,eventos,cont_est,
    polit_invent,coste,interv_v_a)
float
    reloj;
struct estado_sistema
    *estado;
struct lista_eventos
    *eventos;
struct contad_estadisticos
    *cont_est;
struct politica_inventario
    polit_invent;
struct costes
    coste;
struct intervalo
    interv_v_a;
{
    int
        pedido;
    if (estado->inventario < polit_invent.s) {
        pedido=polit_invent.S-estado->inventario;
        if (pedido>0) {
            estado->prod_pedido_suministrador=pedido;
            cont_est->coste_total_pedido+=coste.pedido.gestion+
                pedido*coste.pedido.unidad_producto;
            eventos->t_llegada_producto=reloj+
                Uniforme_interv_A_B(interv_v_a);
        };
    };
    (eventos->t_evaluacion_inventario)++;
    return;
}

void Entrada_datos (polit_invent,t_fin_simulacion)
struct politica_inventario
    *polit_invent;
float
    *t_fin_simulacion;
{
    clrscr();
    printf("\tEntrada de datos:\n");
    printf("\t-----\n");
    printf("\nParametros de gestion del inventario:\n\n\tParametro s: ");
    scanf("%d",&(polit_invent->s));
    printf("\n\tParametro S: ");
    scanf("%d",&(polit_invent->S));
    printf("\nDuracion de la simulacion (meses): ");
    scanf("%f",t_fin_simulacion);
    return;
}

void Informe (reloj,cont_est,coste)
float
    reloj;
struct contad_estadisticos
    cont_est;
struct costes
    coste;
{
    float
        media_pedido,
        media_mantenimiento,
        media_deficit,
        media_coste_total;

    printf("\nFin de la simulacion...\n\n");
    printf("\tI N F O R M E\n");
    printf("\t-----\n\n");
    printf("\tCoste medio mensual de:\n\n");
    media_pedido=cont_est.coste_total_pedido/reloj;
    printf("\tPedidos: \t\t\t%f\n",media_pedido);
    media_mantenimiento=cont_est.integ_prod_inventario*coste.mantenimiento/reloj;
    printf("\tMantenimiento de inventario: \t\t%f\n",media_mantenimiento);
    media_deficit=cont_est.integ_demanda_no_satisfecha*coste.deficit/reloj;
    printf("\tDeficit de inventario: \t\t\t%f\n",media_deficit);
    media_coste_total=media_pedido+media_mantenimiento+media_deficit;
    printf("\tCoste total: \t\t\t%f\n",media_coste_total);
    return;
}

```

```

void Imprime_estado_simulacion(reloj,estado,eventos,cont_est)
float          reloj;
struct estado_sistema      estado;
struct lista_eventos        eventos;
struct contad_estadisticos  cont_est;
{
printf("-----\t\tReloj: %f\n",reloj);
printf("Inventario: %d\t\tProducto encargado al suministrador: %d\n",
      estado.inventario,estado.prod_pedido_suministrador);
printf("Instante ultimo cambio en el inventario: %f\n",
      estado.t_ultimo_cambio_inventario);
printf("Lista de eventos: \n");
printf("Pedido de un cliente: %f\tLlegada de producto: ",
      eventos.t_cliente_pedido);
if (eventos.t_llegada_producto==NO_DEFINIDO)
printf("- \n");
else
printf("%f\n",eventos.t_llegada_producto);
printf("Fin de la simulacion: %f\tEvaluacion inventario: %f\n",
      eventos.t_fin_simulacion,eventos.t_evaluacion_inventario);
printf("Cont. estadisticos:\n");
printf("Coste pedido: %f\tIntegral I+: %f\tIntegral I-: %f\n\n",
      cont_est.coste_total_pedido,cont_est.integ_prod_inventario,
      cont_est.integ_demanda_no_satisfecha);
}

main()
{
struct estado_sistema      estado;
struct lista_eventos        eventos;
struct contad_estadisticos  cont_est;
struct politica_inventario  polit_invent;
struct intervalo            interv_unif;
struct v_a_discreta          unidades_pedidas[4];
struct costes               coste;
float          reloj,
              t_fin_simulacion,
              media_interv_pedidos_cliente;

tipo_evento      event;
Entrada_datos(&polit_invent,&t_fin_simulacion);
Inicializacion(&reloj,&estado,&eventos,&cont_est,t_fin_simulacion,
              &media_interv_pedidos_cliente,&interv_unif,
              unidades_pedidas,&coste);
Imprime_estado_simulacion(reloj,estado,eventos,cont_est);
do {
event=Rutina_tiempo(&reloj,eventos);
if (event==llegada_prod_suministrador)
Evento_llegada_producto(reloj,&estado,&eventos,&cont_est);
else if (event==cliente_realiza_pedido)
Evento_pedido_cliente(reloj,&estado,&eventos,&cont_est,
                      unidades_pedidas,media_interv_pedidos_cliente);
else if (event==fin_simulacion)
Evento_fin_simulacion(reloj,&estado,&cont_est);
else
Evento_evaluacion_inventario(reloj,&estado,&eventos,&cont_est,
                             polit_invent,coste,interv_unif);
Imprime_estado_simulacion(reloj,estado,eventos,cont_est);
}
while (event!=fin_simulacion);
Informe(reloj,cont_est,coste);
return;
}

```

Veamos un ejemplo de la evolución del sistema durante la simulación y el resultado obtenido. El tiempo esta medido en minutos.

```

Entrada de datos:
-----

Parametros de gestion del inventario:

Parametro s: 20
Parametro S: 40

Duracion de la simulacion (meses): 6

-----
Reloj: 0.000000
Inventario: 60      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 0.000000
Lista de eventos:
Pedido de un cliente: 0.454952      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 0.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 0.000000 Integral I-: 0.000000

-----
Reloj: 0.000000
Inventario: 60      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 0.000000
Lista de eventos:
Pedido de un cliente: 0.454952      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 1.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 0.000000 Integral I-: 0.000000

-----
Reloj: 0.454952
Inventario: 58      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 0.454952
Lista de eventos:
Pedido de un cliente: 0.585956      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 1.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 27.297102Integral I-: 0.000000

-----
Reloj: 0.585956
Inventario: 56      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 0.585956
Lista de eventos:
Pedido de un cliente: 0.597448      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 1.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 34.895363Integral I-: 0.000000

-----
Reloj: 0.597448
Inventario: 55      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 0.597448
Lista de eventos:
Pedido de un cliente: 0.975037      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 1.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 35.538929Integral I-: 0.000000

-----
Reloj: 0.975037
Inventario: 52      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 0.975037
Lista de eventos:
Pedido de un cliente: 1.073353      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 1.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 56.306282Integral I-: 0.000000

-----
Reloj: 1.000000
Inventario: 52      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 0.975037
Lista de eventos:
Pedido de un cliente: 1.073353      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 2.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 56.306282Integral I-: 0.000000

```



```

-----
Reloj: 1.073353
Inventario: 48      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 1.073353
Lista de eventos:
Pedido de un cliente: 1.099322      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 2.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 61.418736Integral I-: 0.000000

-----
Reloj: 1.099322
Inventario: 46      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 1.099322
Lista de eventos:
Pedido de un cliente: 1.100583      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 2.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 62.665253Integral I-: 0.000000

-----
Reloj: 1.100583
Inventario: 44      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 1.100583
Lista de eventos:
Pedido de un cliente: 1.125684      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 2.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 62.723248Integral I-: 0.000000

-----
Reloj: 1.125684
Inventario: 40      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 1.125684
Lista de eventos:
Pedido de un cliente: 1.159926      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 2.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 63.827705Integral I-: 0.000000

-----
Reloj: 1.159926
Inventario: 37      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 1.159926
Lista de eventos:
Pedido de un cliente: 1.223698      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 2.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 65.197357Integral I-: 0.000000

-----
Reloj: 1.223698
Inventario: 34      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 1.223698
Lista de eventos:
Pedido de un cliente: 1.413942      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 2.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 67.556938Integral I-: 0.000000

-----
Reloj: 1.413942
Inventario: 31      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 1.413942
Lista de eventos:
Pedido de un cliente: 1.484243      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 2.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 74.025238Integral I-: 0.000000

-----
Reloj: 1.484243
Inventario: 30      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 1.484243
Lista de eventos:
Pedido de un cliente: 1.489574      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 2.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 76.204552Integral I-: 0.000000

-----
Reloj: 1.489574
Inventario: 26      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 1.489574
Lista de eventos:
Pedido de un cliente: 1.511315      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 2.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 76.364487Integral I-: 0.000000

```

```

-----
Reloj: 1.511315
Inventario: 24      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 1.511315
Lista de eventos:
Pedido de un cliente: 1.606694      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 2.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 76.929764Integral I-: 0.000000

-----
Reloj: 1.606694
Inventario: 22      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 1.606694
Lista de eventos:
Pedido de un cliente: 1.981651      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 2.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 79.218857Integral I-: 0.000000

-----
Reloj: 1.981651
Inventario: 20      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 1.981651
Lista de eventos:
Pedido de un cliente: 2.023401      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 2.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 87.467903Integral I-: 0.000000

-----
Reloj: 2.000000
Inventario: 20      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 1.981651
Lista de eventos:
Pedido de un cliente: 2.023401      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 3.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 87.467903Integral I-: 0.000000

-----
Reloj: 2.023401
Inventario: 18      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 2.023401
Lista de eventos:
Pedido de un cliente: 2.046928      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 3.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 88.302902Integral I-: 0.000000

-----
Reloj: 2.046928
Inventario: 15      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 2.046928
Lista de eventos:
Pedido de un cliente: 2.071454      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 3.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 88.726395Integral I-: 0.000000

-----
Reloj: 2.071454
Inventario: 12      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 2.071454
Lista de eventos:
Pedido de un cliente: 2.143805      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 3.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 89.094276Integral I-: 0.000000

-----
Reloj: 2.143805
Inventario: 8       Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 2.143805
Lista de eventos:
Pedido de un cliente: 2.194416      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 3.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 89.962494Integral I-: 0.000000

-----
Reloj: 2.194416
Inventario: 4       Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 2.194416
Lista de eventos:
Pedido de un cliente: 2.230363      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 3.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 90.367386Integral I-: 0.000000

```

```

-----
Reloj: 2.230363
Inventario: 2      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 2.230363
Lista de eventos:
Pedido de un cliente: 2.243052      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 3.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 90.511169Integral I-: 0.000000

-----
Reloj: 2.243052
Inventario: 1      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 2.243052
Lista de eventos:
Pedido de un cliente: 2.282663      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 3.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 90.536545Integral I-: 0.000000

-----
Reloj: 2.282663
Inventario: -3     Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 2.282663
Lista de eventos:
Pedido de un cliente: 2.291721      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 3.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 90.576157Integral I-: 0.000000

-----
Reloj: 2.291721
Inventario: -5     Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 2.291721
Lista de eventos:
Pedido de un cliente: 2.366264      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 3.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 90.576157Integral I-: 0.027173

-----
Reloj: 2.366264
Inventario: -8     Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 2.366264
Lista de eventos:
Pedido de un cliente: 2.406348      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 3.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 90.576157Integral I-: 0.399886

-----
Reloj: 2.406348
Inventario: -9     Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 2.406348
Lista de eventos:
Pedido de un cliente: 2.655068      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 3.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 90.576157Integral I-: 0.720563

-----
Reloj: 2.655068
Inventario: -12    Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 2.655068
Lista de eventos:
Pedido de un cliente: 2.665212      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 3.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 90.576157Integral I-: 2.959045

-----
Reloj: 2.665212
Inventario: -14    Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 2.665212
Lista de eventos:
Pedido de un cliente: 2.860669      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 3.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 90.576157Integral I-: 3.080773

-----
Reloj: 2.860669
Inventario: -18    Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 2.860669
Lista de eventos:
Pedido de un cliente: 3.017327      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 3.000000
Cont. estadisticos:
Coste pedido: 0.000000Integral I+: 90.576157Integral I-: 5.817163

```

```

-----
Reloj: 3.000000
Inventario: -18      Producto encargado al suministrador: 58
Instante ultimo cambio en el inventario: 2.860669
Lista de eventos:
Pedido de un cliente: 3.017327      LLegada de producto: 3.868442
Fin de la simulacion: 6.000000      Evaluacion inventario: 4.000000
Cont. estadisticos:
Coste pedido: 20600.000000      Integral I+: 90.576157 Integral I-: 5.817163

-----
Reloj: 3.017327
Inventario: -21      Producto encargado al suministrador: 58
Instante ultimo cambio en el inventario: 3.017327
Lista de eventos:
Pedido de un cliente: 3.228120      LLegada de producto: 3.868442
Fin de la simulacion: 6.000000      Evaluacion inventario: 4.000000
Cont. estadisticos:
Coste pedido: 20600.000000      Integral I+: 90.576157 Integral I-: 8.637014

-----
Reloj: 3.228120
Inventario: -24      Producto encargado al suministrador: 58
Instante ultimo cambio en el inventario: 3.228120
Lista de eventos:
Pedido de un cliente: 3.299969      LLegada de producto: 3.868442
Fin de la simulacion: 6.000000      Evaluacion inventario: 4.000000
Cont. estadisticos:
Coste pedido: 20600.000000      Integral I+: 90.576157 Integral I-: 13.063653

-----
Reloj: 3.299969
Inventario: -27      Producto encargado al suministrador: 58
Instante ultimo cambio en el inventario: 3.299969
Lista de eventos:
Pedido de un cliente: 3.607379      LLegada de producto: 3.868442
Fin de la simulacion: 6.000000      Evaluacion inventario: 4.000000
Cont. estadisticos:
Coste pedido: 20600.000000      Integral I+: 90.576157 Integral I-: 14.788032

-----
Reloj: 3.607379
Inventario: -31      Producto encargado al suministrador: 58
Instante ultimo cambio en el inventario: 3.607379
Lista de eventos:
Pedido de un cliente: 3.672169      LLegada de producto: 3.868442
Fin de la simulacion: 6.000000      Evaluacion inventario: 4.000000
Cont. estadisticos:
Coste pedido: 20600.000000      Integral I+: 90.576157 Integral I-: 23.088121

-----
Reloj: 3.672169
Inventario: -34      Producto encargado al suministrador: 58
Instante ultimo cambio en el inventario: 3.672169
Lista de eventos:
Pedido de un cliente: 3.677849      LLegada de producto: 3.868442
Fin de la simulacion: 6.000000      Evaluacion inventario: 4.000000
Cont. estadisticos:
Coste pedido: 20600.000000      Integral I+: 90.576157 Integral I-: 25.096598

-----
Reloj: 3.677849
Inventario: -36      Producto encargado al suministrador: 58
Instante ultimo cambio en el inventario: 3.677849
Lista de eventos:
Pedido de un cliente: 3.879169      LLegada de producto: 3.868442
Fin de la simulacion: 6.000000      Evaluacion inventario: 4.000000
Cont. estadisticos:
Coste pedido: 20600.000000      Integral I+: 90.576157 Integral I-: 25.289703

-----
Reloj: 3.868442
Inventario: 22      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 3.868442
Lista de eventos:
Pedido de un cliente: 3.879169      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 4.000000
Cont. estadisticos:
Coste pedido: 20600.000000      Integral I+: 90.576157 Integral I-: 32.151070

-----
Reloj: 3.879169
Inventario: 21      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 3.879169
Lista de eventos:
Pedido de un cliente: 3.925141      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 4.000000
Cont. estadisticos:
Coste pedido: 20600.000000      Integral I+: 90.812157 Integral I-: 32.151070

```

```

-----
Reloj: 3.925141
Inventario: 19      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 3.925141
Lista de eventos:
Pedido de un cliente: 3.997228      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 4.000000
Cont. estadisticos:
Coste pedido: 20600.000000      Integral I+: 91.777550 Integral I-: 32.151070

-----
Reloj: 3.997228
Inventario: 17      Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 3.997228
Lista de eventos:
Pedido de un cliente: 4.014430      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 4.000000
Cont. estadisticos:
Coste pedido: 20600.000000      Integral I+: 93.147209 Integral I-: 32.151070

-----
Reloj: 4.000000
Inventario: 17      Producto encargado al suministrador: 23
Instante ultimo cambio en el inventario: 3.997228
Lista de eventos:
Pedido de un cliente: 4.014430      LLegada de producto: 4.646240
Fin de la simulacion: 6.000000      Evaluacion inventario: 5.000000
Cont. estadisticos:
Coste pedido: 30700.000000      Integral I+: 93.147209 Integral I-: 32.151070

-----
Reloj: 4.014430
Inventario: 14      Producto encargado al suministrador: 23
Instante ultimo cambio en el inventario: 4.014430
Lista de eventos:
Pedido de un cliente: 4.090752      LLegada de producto: 4.646240
Fin de la simulacion: 6.000000      Evaluacion inventario: 5.000000
Cont. estadisticos:
Coste pedido: 30700.000000      Integral I+: 93.439636 Integral I-: 32.151070

-----
Reloj: 4.090752
Inventario: 13      Producto encargado al suministrador: 23
Instante ultimo cambio en el inventario: 4.090752
Lista de eventos:
Pedido de un cliente: 4.156806      LLegada de producto: 4.646240
Fin de la simulacion: 6.000000      Evaluacion inventario: 5.000000
Cont. estadisticos:
Coste pedido: 30700.000000      Integral I+: 94.508148 Integral I-: 32.151070

-----
Reloj: 4.156806
Inventario: 10      Producto encargado al suministrador: 23
Instante ultimo cambio en el inventario: 4.156806
Lista de eventos:
Pedido de un cliente: 4.229056      LLegada de producto: 4.646240
Fin de la simulacion: 6.000000      Evaluacion inventario: 5.000000
Cont. estadisticos:
Coste pedido: 30700.000000      Integral I+: 95.366844 Integral I-: 32.151070

-----
Reloj: 4.229056
Inventario: 7      Producto encargado al suministrador: 23
Instante ultimo cambio en el inventario: 4.229056
Lista de eventos:
Pedido de un cliente: 4.497318      LLegada de producto: 4.646240
Fin de la simulacion: 6.000000      Evaluacion inventario: 5.000000
Cont. estadisticos:
Coste pedido: 30700.000000      Integral I+: 96.089355 Integral I-: 32.151070

-----
Reloj: 4.497318
Inventario: 4      Producto encargado al suministrador: 23
Instante ultimo cambio en el inventario: 4.497318
Lista de eventos:
Pedido de un cliente: 4.615910      LLegada de producto: 4.646240
Fin de la simulacion: 6.000000      Evaluacion inventario: 5.000000
Cont. estadisticos:
Coste pedido: 30700.000000      Integral I+: 97.967186 Integral I-: 32.151070

-----
Reloj: 4.615910
Inventario: 2      Producto encargado al suministrador: 23
Instante ultimo cambio en el inventario: 4.615910
Lista de eventos:
Pedido de un cliente: 4.639556      LLegada de producto: 4.646240
Fin de la simulacion: 6.000000      Evaluacion inventario: 5.000000
Cont. estadisticos:
Coste pedido: 30700.000000      Integral I+: 98.441551 Integral I-: 32.151070

```

```

-----
Reloj: 4.639556
Inventario: 1      Producto encargado al suministrador: 23
Instante ultimo cambio en el inventario: 4.639556
Lista de eventos:
Pedido de un cliente: 4.676909      LLegada de producto: 4.646240
Fin de la simulacion: 6.000000      Evaluacion inventario: 5.000000
Cont. estadisticos:
Coste pedido: 30700.000000      Integral I+: 98.488846 Integral I-: 32.151070

-----
Reloj: 4.646240
Inventario: 24     Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 4.646240
Lista de eventos:
Pedido de un cliente: 4.676909      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 5.000000
Cont. estadisticos:
Coste pedido: 30700.000000      Integral I+: 98.495529 Integral I-: 32.151070

-----
Reloj: 4.676909
Inventario: 22     Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 4.676909
Lista de eventos:
Pedido de un cliente: 4.691369      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 5.000000
Cont. estadisticos:
Coste pedido: 30700.000000      Integral I+: 99.231590 Integral I-: 32.151070

-----
Reloj: 4.691369
Inventario: 19     Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 4.691369
Lista de eventos:
Pedido de un cliente: 4.700537      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 5.000000
Cont. estadisticos:
Coste pedido: 30700.000000      Integral I+: 99.549698 Integral I-: 32.151070

-----
Reloj: 4.700537
Inventario: 17     Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 4.700537
Lista de eventos:
Pedido de un cliente: 4.744127      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 5.000000
Cont. estadisticos:
Coste pedido: 30700.000000      Integral I+: 99.723892 Integral I-: 32.151070

-----
Reloj: 4.744127
Inventario: 15     Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 4.744127
Lista de eventos:
Pedido de un cliente: 4.963447      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 5.000000
Cont. estadisticos:
Coste pedido: 30700.000000      Integral I+: 100.464920      Integral I-: 32.151070

-----
Reloj: 4.963447
Inventario: 12     Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 4.963447
Lista de eventos:
Pedido de un cliente: 5.241070      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 5.000000
Cont. estadisticos:
Coste pedido: 30700.000000      Integral I+: 103.754715      Integral I-: 32.151070

-----
Reloj: 5.000000
Inventario: 12     Producto encargado al suministrador: 28
Instante ultimo cambio en el inventario: 4.963447
Lista de eventos:
Pedido de un cliente: 5.241070      LLegada de producto: 5.759190
Fin de la simulacion: 6.000000      Evaluacion inventario: 6.000000
Cont. estadisticos:
Coste pedido: 42300.000000      Integral I+: 103.754715      Integral I-: 32.151070

-----
Reloj: 5.241070
Inventario: 8      Producto encargado al suministrador: 28
Instante ultimo cambio en el inventario: 5.241070
Lista de eventos:
Pedido de un cliente: 5.398213      LLegada de producto: 5.759190
Fin de la simulacion: 6.000000      Evaluacion inventario: 6.000000
Cont. estadisticos:
Coste pedido: 42300.000000      Integral I+: 107.086197      Integral I-: 32.151070

```

```

-----
Reloj: 5.398213
Inventario: 5      Producto encargado al suministrador: 28
Instante ultimo cambio en el inventario: 5.398213
Lista de eventos:
Pedido de un cliente: 5.450909      LLegada de producto: 5.759190
Fin de la simulacion: 6.000000      Evaluacion inventario: 6.000000
Cont. estadisticos:
Coste pedido: 42300.000000      Integral I+: 108.343338      Integral I-: 32.151070

```

```

-----
Reloj: 5.450909
Inventario: 3      Producto encargado al suministrador: 28
Instante ultimo cambio en el inventario: 5.450909
Lista de eventos:
Pedido de un cliente: 5.572228      LLegada de producto: 5.759190
Fin de la simulacion: 6.000000      Evaluacion inventario: 6.000000
Cont. estadisticos:
Coste pedido: 42300.000000      Integral I+: 108.606819      Integral I-: 32.151070

```

```

-----
Reloj: 5.572228
Inventario: 1      Producto encargado al suministrador: 28
Instante ultimo cambio en el inventario: 5.572228
Lista de eventos:
Pedido de un cliente: 5.843253      LLegada de producto: 5.759190
Fin de la simulacion: 6.000000      Evaluacion inventario: 6.000000
Cont. estadisticos:
Coste pedido: 42300.000000      Integral I+: 108.970779      Integral I-: 32.151070

```

```

-----
Reloj: 5.759190
Inventario: 29     Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 5.759190
Lista de eventos:
Pedido de un cliente: 5.843253      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 6.000000
Cont. estadisticos:
Coste pedido: 42300.000000      Integral I+: 109.157738      Integral I-: 32.151070

```

```

-----
Reloj: 5.843253
Inventario: 26     Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 5.843253
Lista de eventos:
Pedido de un cliente: 6.045033      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 6.000000
Cont. estadisticos:
Coste pedido: 42300.000000      Integral I+: 111.595581      Integral I-: 32.151070

```

```

-----
Reloj: 6.000000
Inventario: 26     Producto encargado al suministrador: 0
Instante ultimo cambio en el inventario: 6.000000
Lista de eventos:
Pedido de un cliente: 6.045033      LLegada de producto: -
Fin de la simulacion: 6.000000      Evaluacion inventario: 6.000000
Cont. estadisticos:
Coste pedido: 42300.000000      Integral I+: 115.670998      Integral I-: 32.151070

```

Fin de la simulacion...

I N F O R M E

Coste medio mensual de:

Pedidos:	7050.000000
Mantenimiento de inventario:	1927.849976
Deficit de inventario:	2679.255859
Coste total:	11657.105469

El objetivo de la modelización y simulación de este sistema es comparar las nueve siguientes políticas de gestión del inventario.

s	20	20	20	20	40	40	40	60	60
S	40	60	80	100	60	80	100	80	100

Para ello vamos a simular 5 veces cada una (obsérvese la arbitrariedad en la elección), con un tiempo de simulación de 120 meses, para determinar cuál es la que produce un coste total medio mensual estimado menor para los primeros 120 meses. El coste total medio mensual es la suma del coste medio mensual de los pedidos al suministrador, el coste medio mensual del mantenimiento del inventario y el coste medio mensual de la demanda del cliente que no puede ser satisfecha por falta de inventario.

A continuación se dan los resultados obtenidos. En las tablas de resultados se ha seguido la siguiente notación:

Simulación: número de simulación.
 Pedidos: coste medio mensual de los pedidos al suministrador.
 Manten.: coste medio mensual del mantenimiento del inventario.
 Déficit: coste medio mensual de la demanda del cliente que no puede ser satisfecha.

En la última columna de la tabla se dan los valores medios de las 5 simulaciones.

(s=20,S=40). Tiempo simulación = 120 meses

Simulación	1	2	3	4	5	Media
Pedidos	9952.5	10345.8	10174.2	9876.7	9755.8	10021.0
Manten.	910.2	841.1	845.6	888.0	857.4	868.5
Déficit	1938.0	2064.2	2180.8	2184.8	1944.5	2062.5
Total	12810.7	13251.1	13200.6	12949.4	12557.6	12953.8

(s=20,S=60). Tiempo simulación = 120 meses

Simulación	1	2	3	4	5	Media
Pedidos	8885.0	8797.5	9022.5	8712.5	8707.5	8825.0
Manten.	1680.0	1788.4	1711.8	1761.3	1920.9	1772.5
Déficit	1444.5	1366.3	1562.5	1090.2	1254.1	1343.5
Total	12009.5	11952.2	12296.8	11564.1	11882.5	11941.0

(s=20,S=80). Tiempo simulación = 120 meses

Simulación	1	2	3	4	5	Media
Pedidos	7946.7	8604.2	8536.7	8606.7	8020.0	8342.9
Manten.	2855.8	2628.9	2709.6	2541.4	2867.4	2720.6
Déficit	896.4	1116.0	1080.3	1133.9	653.4	976.0
Total	11698.9	12349.1	12326.5	12282.0	11540.9	12039.5

(s=20,S=100). Tiempo simulación = 120 meses

Simulación	1	2	3	4	5	Media
Pedidos	8184.2	8144.2	8646.7	8116.7	8909.2	8400.2
Manten.	3700.4	3602.7	3621.1	3778.1	3642.3	3668.9
Déficit	647.8	815.3	666.9	592.4	836.0	711.7
Total	12532.4	12562.2	12934.6	12487.2	13387.4	12780.8

(s=40,S=60). Tiempo simulación = 120 meses

Simulación	1	2	3	4	5	Media
Pedidos	9886.7	10053.3	10120.0	9791.7	9619.2	9894.2
Manten.	2392.2	2556.0	2523.1	2574.6	2574.7	2524.1
Déficit	367.6	201.6	197.8	120.9	272.8	232.1
Total	12642.5	12811.0	12840.9	12487.1	12466.7	12649.6

(s=40,S=80). Tiempo simulación = 120 meses

Simulación	1	2	3	4	5	Media
Pedidos	8432.5	8990.0	8850.0	9025.0	8950.0	8849.5
Manten.	3556.3	3466.8	3508.6	3505.4	3422.4	3491.9
Déficit	136.6	225.6	145.6	124.8	188.0	164.1
Total	12125.4	12682.3	12504.1	12655.3	12560.5	12505.5

(s=40,S=100). Tiempo simulación = 120 meses

Simulación	1	2	3	4	5	Media
Pedidos	8576.7	8515.0	8578.3	9139.2	8155.0	8592.8
Manten.	4351.8	4488.1	4498.0	4314.5	4523.5	4435.2
Déficit	167.1	141.3	143.8	78.3	58.0	117.7
Total	13095.6	13144.4	13220.1	13532.0	12736.6	13145.5

(s=60,S=80). Tiempo simulación = 120 meses

Simulación	1	2	3	4	5	Media
Pedidos	10691.7	10075.8	9952.5	9621.7	9793.3	10027.0
Manten.	4316.9	4462.7	4498.0	4495.6	4545.6	4463.8
Déficit	6.9	18.4	11.3	5.3	0.0	8.4
Total	15015.5	14556.8	14461.8	14122.5	14338.9	14499.1

(s=60,S=100). Tiempo simulación = 120 meses

Simulación	1	2	3	4	5	Media
Pedidos	8575.8	9135.8	8946.7	8624.2	8842.5	8825.0
Manten.	5658.5	5436.5	5485.8	5646.7	5631.8	5571.9
Déficit	0.0	5.3	0.1	1.0	0.5	1.4
Total	14234.3	14577.6	14432.6	14271.9	14474.8	14398.2

Los resultados obtenidos parecen indicar que la política (20,60) es la preferible. Sin embargo, cabe preguntarse, ¿hasta qué punto son fiables los resultados obtenidos?, ¿En qué medida influye el número de replicaciones de la simulación, que arbitrariamente hemos fijado en 5, en el resultado?. Como vemos, hasta que no respondamos a estas preguntas no podremos afirmar, con un tanto por ciento de confianza, que una determinada política es la mejor.

1.4- Simulación de Monte Carlo: cálculo de una integral definida.

Bajo el nombre genérico de "simulaciones de Monte Carlo" suelen englobarse todas las simulaciones que emplean números aleatorios para resolver problemas estáticos (estocásticos o deterministas), es decir, problemas en los cuales el tiempo no juega un papel relevante.

Supongamos, por ejemplo, que queremos evaluar la integral

$$I = \int_a^b g(x) dx$$

donde $g(x)$ es una función real monovaluada. Resolvamos este problema determinista mediante simulación de Monte Carlo.

Sea Y una variable aleatoria definida de la forma:

$$Y = (b - a)g(X)$$

donde X es una variable aleatoria uniforme en el intervalo $[a, b]$, $U(a, b)$.

El valor esperado de Y , $E(Y)$, es:

$$E(Y) = E((b - a)g(X)) = (b - a)E(g(X)) = (b - a) \int_a^b g(x) f_x(x) dx = (b - a) \frac{\int_a^b g(x) dx}{b - a} = I$$

donde la función $f_x(x) = \frac{1}{b - a}$ es la función densidad de probabilidad de la variable aleatoria uniforme en el intervalo $[a, b]$, $U(a, b)$.

Así pues, el problema de evaluar la integral se ha reducido a la estima del valor esperado $E(Y)$.

En particular, $E(Y)$ puede estimarse calculando la media de cierto número, n , de muestras:

$$E(Y)(n) = \frac{\sum_{i=1}^n Y_i}{n} = (b - a) \frac{\sum_{i=1}^n g(X_i)}{n}$$

donde X_1, \dots, X_n son variables aleatorias $U(a, b)$ IID.

Puede demostrarse que $E(E(Y)(n)) = I$, es decir, que $E(Y)(n)$ es un estimador no sesgado de I , y $Var(E(Y)(n)) = \frac{Var(Y)}{n}$. Como $Var(Y)$ es un número fijo, $E(Y)(n)$ puede hacerse arbitrariamente próximo a I para un n suficientemente grande.

Como ejemplo, para ilustrar lo anteriormente dicho, supongamos que deseamos evaluar la integral

$$\int_0^\pi \sin(x) dx$$

La integral posee primitiva analítica ($-\cos(x)$), y el valor de la integral definida entre 0 y π es 2.

En la práctica, si el integrando tiene "buen comportamiento" (ni la función ni sus derivadas presentan discontinuidades) suelen ser preferibles otros métodos numéricos de integración al de Monte Carlo.

El siguiente programa realiza la simulación:

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>

#define PI 3.141593

struct intervalo {
    float inicio,
          fin;
};

float Uniforme_interv_A_B(interv)
struct intervalo interv;
{
    /* Generacion de U(A,B) */
    float v_unif;
    v_unif=rand()/32727.0;
    if (v_unif>1.0) v_unif=1.0;
    return(interv.inicio+(interv.fin-interv.inicio)*v_unif);
}

float Func_integrando(x)
float x;
{
    return(sin(x));
}

void Entrada_datos(num_muestras,interv_integ)
unsigned int *num_muestras;
struct intervalo *interv_integ;
{
    clrscr();
    /* Numero de muestras */
    printf("\tNumero de muestras (1-65535): ");
    scanf("%d",num_muestras);
    /* Intervalo de integracion */
    interv_integ->inicio=0;
    interv_integ->fin=PI;
    /* Inicializacion rutina generacion numeros aleatorios */
    randomize();
    return;
}

void Informe(num_muestras,estima_integ)
unsigned int num_muestras;
float estima_integ;
{
    printf("\tI N F O R M E\n\t-----\n\n");
    printf("Numero de muestras: %d\n",num_muestras);
    printf("Estima de la integral: %f\n",estima_integ);
}

main()
{
    unsigned int num_muestras,
                 contad=0;
    float func_acumulada=0,
          estima_integ;
    struct intervalo interv_integ;
    Entrada_datos(&num_muestras,&interv_integ);
    while (contad<num_muestras) {
        func_acumulada+=Func_integrando(Uniforme_interv_A_B(interv_integ));
        contad++;
    }
}
```

```

    }
    estima_integ=(interv_integ.fin-interv_integ.inicio)*
        func_acumulada/num_muestras;
    Informe(num_muestras,estima_integ);
    return;
}

```

Simulando varias veces, para diferentes valores del número de muestras, obtenemos los siguientes resultados.

Simulación 1:

n	10	50	100	500	1000
E(Y)(n)	1.980	1.832	2.021	1.958	2.001

Simulación 2:

n	10	50	100	500	1000
E(Y)(n)	2.302	1.845	1.799	1.971	1.993

Simulación 3:

n	10	50	100	500	1000
E(Y)(n)	2.286	1.926	1.801	1.999	1.998

Simulación 4:

n	10	50	100	500	1000
E(Y)(n)	2.371	1.867	1.926	1.943	1.937

Simulación 5:

n	10	50	100	500	1000
E(Y)(n)	2.300	1.968	1.916	1.975	2.004

Como vemos, deben establecerse criterios para determinar el tamaño de la muestra, n, que permita obtener una determinada precisión.

Similarmente si queremos evaluar la integral: $I = \int_0^{\infty} g(x) dx$

podemos realizar el cambio de variable: $y = \frac{1}{x+1} \rightarrow dy = \frac{-dx}{(x+1)^2} = -y^2 dx$

con lo cual, reescribimos la integral: $I = \int_0^{\infty} g(x) dx = \int_0^1 \frac{g\left(\frac{1}{y}-1\right)}{y^2} dy$

1.5- Ejercicio. Varios empleados, una cola por empleado y posibilidad de que los clientes cambien de cola.

Se propone al alumno la simulación de una sucursal bancaria atendida por 5 empleados, cada uno con su propia cola de espera de clientes. Los clientes pueden ser atendidos indistintamente por cualquiera de los empleados y pueden cambiarse de cola a otra más corta. Cuando el empleado acaba de atender a un cliente, éste se marcha.

El banco abre sus puertas a las 9'00 horas y las cierra a las 17'00, continuando el servicio hasta que todos los clientes, que en el momento de cierre se encontraban en el banco, son atendidos.

Supongamos que los clientes llegan de uno en uno, que el tiempo entre llegadas sucesivas de clientes es una variable aleatoria exponencial IID de media 1 minuto y que el tiempo que el empleado tarda en atender a un cliente (desde que comienza a atenderle hasta que se marcha) es una variable aleatoria exponencial IID de media 4.5 minutos.

Cada empleado tiene su propia cola. El cliente al llegar se pone al final de la cola más corta. En el caso de que varias colas tengan la longitud mínima, se pondrá en la situada más a la izquierda. Es decir, si las colas se numeran de izquierda a derecha, se situará, en caso de igualdad, al final de la cola cuya numeración sea menor.

Sea n_j el número total de clientes (el que está siendo atendido más los que esperan) frente al empleado j -ésimo, en un instante determinado. Si se produce la situación en que dos colas, i, j , satisfacen $n_j > n_i + 1$, entonces el último cliente de la cola j se cambia a la cola i . Si la situación se da para varias colas j y una misma i , se cambia el cliente de la cola j más próxima y, en caso de igualdad, el de la situada a la izquierda de la cola i . Si el empleado que atiende la cola i esta libre, el cliente procedente de la cola j es atendido inmediatamente.

El director del banco esta interesado en analizar la calidad del servicio que se presta actualmente, con 5 empleados, y del que se prestaría si se contrataran 1 ó 2 empleados más. Para ello, se desea conocer el número medio de clientes en las colas y el tiempo medio y máximo de espera en la cola por cliente. En todos los casos ($n=5,6,7$), suponemos que en el instante de apertura del banco no hay ningún cliente esperando.

El modelo posee 3 tipos de eventos:

- Llegada de un nuevo cliente al banco. Debe estimarse el instante en que se producirá la siguiente llegada. Si hay algún empleado libre, atiende inmediatamente al recién llegado, estimándose el instante en que se producirá su marcha. Si todos los empleados están ocupados, el cliente se pone al final de la cola más corta.
- Marcha de un cliente una vez finalizado su servicio. Si no hay ningún cliente en su cola, el empleado queda libre. En caso contrario, el primero de la cola es atendido, estimándose el instante en que se producirá su marcha. En cualquier caso, se analiza si la nueva situación de la cola aconseja que algún cliente de otra cola se cambie a ésta. En caso afirmativo, debe darse de baja al cliente en su antigua cola y, dependiendo de si el empleado de la nueva cola está libre o no, será atendido inmediatamente y estimado su instante de marcha o deberá esperar su turno al final de la cola.
- Cierre de las puertas del banco a las 17'00 horas. La simulación continua hasta que todos los clientes de las colas han sido atendidos y se han marchado.

Deben especificarse los valores de los siguientes parámetros de entrada por teclado:

- Número de horas que el banco abre sus puertas. Si el horario es de 9'00 a 17'00, abre durante 8 horas.
- Medias de las variables aleatorias exponenciales (intervalo entre llegadas y tiempo de servicio).
- Número de empleados de la sucursal.

El objetivo de la simulación es estimar:

- Tiempo medio de espera en la cola por cliente.
- Número promedio total de clientes en las colas.
- Tiempo máximo de espera en la cola por cliente.

para los casos que la sucursal esté atendida por 5, 6 y 7 empleados. Repetir 10 veces cada simulación y, a la vista de los valores medios obtenidos, decidir si compensa (es una apreciación subjetiva) la contratación de uno o dos empleados más.

Contadores estadísticos:

- Número total de clientes que han comenzado a ser atendidos. Es necesario para calcular el tiempo medio de espera en la cola, dividiendo la suma de los tiempos de espera de todos los clientes atendidos por el número de éstos.
- La suma del tiempo de espera de todos los clientes que han comenzado a ser atendidos. Para calcular el tiempo de espera de un cliente debe conocerse el instante en que llegó y restarlo del instante en que comienza a ser atendido.
- La integral temporal del número de clientes que hay en las colas. Para ello, teniendo en cuenta que la función "número de clientes en las colas" es una función entera discontinua en el tiempo, cada vez que llega o se marcha un cliente, se calcula la contribución a la integral del intervalo entre el evento anterior y éste, como el producto de la longitud del intervalo temporal por el número total de clientes en las colas durante él. Para ello, se guarda el número total de personas en las colas y el instante en que ocurrió el evento anterior.

El lenguaje de programación a emplear puede ser elegido libremente por el alumno, con la única restricción de que debe permitir gestionar dinámicamente la memoria (C, Pascal, etc.). Debe cuidarse el estilo de programación, que deberá ser estructurado, y deberá sacarse partido de la gestión dinámica de memoria.

Debe entregarse:

- Una memoria escrita con:

Diagramas de flujo de las rutinas del programa y del programa principal. Explicación de la función de cada rutina.

Resultados obtenidos en las 10 simulaciones realizadas para cada número de empleados (5, 6 y 7) y valoración (subjetiva) de si compensa la ampliación de plantilla.

- Un disquete con:

Fuente del programa, ampliamente documentado.

Tres ficheros con la evolución, a lo largo de una simulación (una cualquiera de las 10 repeticiones), del estado del sistema, contadores estadísticos, lista de eventos y reloj de la simulación. Un fichero para el caso con 5 empleados, otro con 6 y otro con 7.