Proyecto Final Simulación

Buttignol Leandro y López Martín

17 de junio 2016

1 Consigna

Consideremos un ascensor que se puede comandar por los siguientes eventos: arriba,abajo,parar. Cada piso a su vez cuenta con un sensor que indica la presencia del ascensor, de modo tal que la salida del sistema "ascensor" son los eventos producidos por dichos sensores.

Algunas consideraciones:
• El ascensor sube y baja a velocidades contantes de 1 metro por segundo.
• La distancia entre un piso y otro es de 2 metros.
• El edificio tiene 10 pisos.

1.1 Extensión 1

Diseñar un modelo de un controlador para el ascensor al cual llegan eventos con números enteros, indicando que el ascensor debe ir hasta dicho piso. Dicho controlador recibe además como eventos de entrada, las salidas del ascensor. El sistema debe provocar una señal de salida "libre" cada vez que finalice una tarea y "ocupado" cada vez que comience otra. Suponer que no arriban eventos de entrada indicando que se debe ir a un nuevo piso mientras se está yendo a otro. Los pisos a los cual debe dirigirse el ascensor son igualmente probables.

1.2 Extensión 2

Obtener el modelo del tablero del ascensor, suponiendo que consta de una cola en la que almacena pedidos (observar que no debe haber dos pedidos consecutivos iguales, ya que no tiene sentido ir dos veces al mismo lugar). Los arribos de los pedidos al tablero siguen una distribución exponencial con una media de 1 cada 9 segundos.

1.3 Extensión 3

Suponiendo que se administran 2 ascensores, diseñar un modelo para el control de pedidos desde los pisos. Suponer que al mismo llegan pedidos del tipo "ir al piso j" que indican que desde el piso j se pidió el ascensor. Dichos pedidos pueden ser enviados a uno u otro tablero de ascensor según alguna estrategia adoptada.

1.4 Primer entrega

• Diseñe un modelo preliminar del problema.
• Verifique si las colas detectadas son estables.
• Estimar analíticamente el tiempo medio que una solicitud pasa en la cola (tiempo de respuesta).
• Especifique su comportamiento mediente el formalismo DEVS.

2 Devs atómicos.

2.1 Dev de ascensor.

AscensorDev $\langle X, Y, S, \delta int, \delta ext, \lambda, ta\rangle$

$X = \{"Arriba", "Abajo", "Parar"\}$
$Y = \{0..9\}$
$S = \{\{0..9\} \times R^{+} \times \{ "Arriba", "Abajo", "Parar"\}\}$

El estado S está conformado por el piso actual$\times$*tiempo* $\times$ *acción*.

$\delta ext(s, e, x) = \delta ext((piso, tiempo, accion), e, x)$

$$\begin{cases} (piso, \infty, x) & \text{si accion="Parar" } \wedge \text{ x="Parar} \\ (piso + 1, \frac{alturaPiso}{velocidad}, x) & \text{si accion="Parar" } \wedge \text{ x="Arriba"} \\ (piso - 1, \frac{alturaPiso}{velocidad}, x) & \text{si accion="Parar" } \wedge \text{ x="Abajo"} \\ (piso, tiempo - e, accion) & \text{si (accion="Abajo" } \vee \text{ accion="Arriba) } \wedge \text{ (x="Abajo" } \vee \text{ x="Arriba)} \\ (piso, \infty, x) & \text{si ((accion="Abajo"}\vee accion = "Arriba) \wedge x = "Parar") \end{cases}$$

$\delta int(s) = \delta int(piso, tiempo, accion)$

$$\begin{cases} (piso, \infty, "Parar") & \text{si accion="Parar"} \\ (piso + 1, \frac{alturaPiso}{velocidad}, accion) & \text{si accion="Arriba"} \\ (piso - 1, \frac{alturaPiso}{velocidad}, accion) & \text{si accion="Abajo"} \end{cases}$$

$\lambda(s) = \lambda(piso, tiempo, accion) = piso$

$ta(s) = ta(piso, tiempo, accion) = tiempo$

$SInicial = (0, \infty, "Parar")$

2.2 Dev de controlador de 1 ascensor.

Controlador1Dev$\langle X, Y, S, \delta int, \delta ext, \lambda, ta\rangle$

$X = \{0..9\} \times \{0, 1\}$
$Y = \{\,"Arriba",\ "Abajo",\ "Parar"\} \times \{0\} \cup \{"Libre","Ocupado"\} \times \{1\}$
$S = \{\{0..9\} \times \{0..9\} \times R^{+} \times \{"Libre","Ocupado"\} \times \{0..9\}\}$

El estado S está conformado por piso actual $\times$ destino $\times$ tiempo $\times$ estado del ascensor $\times$ puerto por el que será la prox. salida.

δext(s,e,x)=δext((piso,destino,tiempo,estado,salida),e,(n,p))
$$\begin{cases} (piso, n, 0, "Ocupado", salida) & \text{si estado="Libre"}\wedge p = 0 \\ (n, destino, 0, estado, 0) & \text{si estado="Ocupado"}\wedge p = 1 \wedge n = destino \\ (n, destino, \infty, estado, 0) & \text{estado="Ocupado"}\wedge p = 1 \wedge n \neq destino \end{cases}$$

$\delta int(s) = \delta int(piso, destino, tiempo, estado, salida)$
$$\begin{cases} (piso, destino, 0, "Libre", 1) & \text{si salida=0 } \wedge estado = "Ocupado" \wedge piso = destino \\ (piso, destino, 0, estado, 1) & \text{si salida=0 } \wedge estado = "Ocupado" \wedge piso \neq destino \\ (piso, destino, \infty, estado, 0) & \text{salida=1} \end{cases}$$

$\lambda(s) = \lambda(piso, destino, tiempo, estado, salida)$
$$\begin{cases} ("Parar", salida) & \text{si salida=0 } \wedge piso = destino \\ ("Abajo", salida) & \text{si salida=0 } \wedge piso > destino \\ ("Arriba", salida) & \text{si salida=0 } \wedge piso < destino \\ (estado, salida) & \text{si salida=1} \end{cases}$$

$ta(s) = ta(piso, destino, tiempo, estado, salida) = tiempo$

$SInicial = (0, 0, \infty, "Libre", 0)$

2.3 Dev de controlador de 2 ascensores

Controlador2Dev$\langle X, Y, S, \delta int, \delta ext, \lambda, ta\rangle$

$X = \{0..9\} \times \{0, 1, 2\}$
$Y = \{"Arriba", "Abajo", "Parar"\} \times \{0, 1\} \cup \{"Libre", "Ocupado"\} \times \{2\}$
$S = (a1, a2, \{0, 1, 2\})$
$S = ((\{0..9\}\times\{0..9\}\times\{"Libre", "Ocupado"\}\times R^+)\times(\{0..9\}\times\{0..9\}\times\{"Libre", "Ocupado"\}\times R^+))\times\{0, 1, 2\})$

El estado S está compuesto por a1 × a2 × puerto de la prox.salida. Donde a1 y a2 son los estados de ascensores distintos. Dichos estados están compuesto por piso actual × piso destino × estado × tiempo.

$$\delta ext((a1, a2, sal), e, (n, p)) = \delta ext((p1, d1, e1, t1), (p2, d2, e2, t2), sal), e, (n, p))$$
$$\begin{cases} ((p1, n, "Ocupado", 0), (p2, d2, e2, t2 - e), 0) & \text{estado(a1,a2)="Libre"} \wedge p = 0 \wedge optimo(a1, a2, n) = "a1" \\ ((p1, d1, e1, t1 - e), (p2, n, "Ocupado", 0), 1) & \text{estado(a1,a2)="Libre"} \wedge p = 0 \wedge optimo(a1, a2, n) = "a2" \\ ((n, d1, e1, 0), (p2, d2, e2, t2 - e), 0) & \text{estado(a1,a2)="Ocupado"} \wedge p = 1 \wedge n = d1 \\ ((n, d1, e1, \infty), (p2, d2, e2, t2 - e), 1) & \text{estado(a1,a2)="Ocupado"} \wedge p = 1 \wedge n \neq d1 \\ ((p1, d1, e1, t1 - e), (n, d2, e2, 0), 1) & \text{estado(a1,a2)="Ocupado"} \wedge p = 2 \wedge n = d2 \\ ((p1, d1, e1, t1 - e), (n, d2, e2, \infty), 0) & \text{estado(a1,a2)="Ocupado"} \wedge p = 2 \wedge n \neq d2 \end{cases}$$

$$\delta int(a1, a2, sal) = \delta int((p1, d1, e1, t1), (p2, d2, e2, t2), sal)$$
$$\begin{cases} ((p1, d1, "Libre", 0), a2, 2) & \text{si sal=0} \wedge e1 = "Ocupado" \wedge t1 \leq t2 \wedge p1 = d1 \\ ((p1, d1, e1, 0), a2, 2) & \text{sal=0} \wedge e1 = "Ocupado" \wedge t1 \leq t2 \wedge p1 \neq d1 \\ (a1, (p2, d2, "Libre", 0), 2) & \text{sal=1} \wedge e2 = "Ocupado" \wedge t1 > t2 \wedge p2 = d2 \\ (a1, (p2, d2, e2, 0), 2) & \text{sal=1} \wedge e2 = "Ocupado" \wedge t1 > t2 \wedge p2 \neq d2 \\ ((p1, d1, e1, \infty), a2, 1) & \text{sal=2} \wedge t1 = 0 \\ (a1, (p2, d2, e2, \infty), 0) & \text{sal=2} \wedge t2 = 0 \end{cases}$$

$$\lambda(a1, a2, sal) = \lambda((p1, d1, e1, t1), (p2, d2, e2, t2), sal)$$
$$\begin{cases} ("Parar", sal) & \text{si sal=0} \wedge p1 = d1 \\ ("Abajo", sal) & \text{si sal=0} \wedge p1 > d1 \\ ("Arriba", sal) & \text{si sal=0} \wedge p1 < d1 \\ ("Parar", sal) & \text{si sal=1} \wedge p2 = d2 \\ ("Abajo", sal) & \text{si sal=1} \wedge p2 > d2 \\ ("Arriba", sal) & \text{si sal=1} \wedge p2 < d2 \\ (estado(a1, a2), sal) & \text{si sal=2} \end{cases}$$

$ta(a1, a2, sal) = ta((p1, d1, e1, t1), (p2, d2, e2, t2), sal) = Min(t1, t2)$

$SInicial((0, 0, "Libre", \infty), (0, 0, "Libre", \infty), 1)$

Donde,

-estado(a1,a2) es una función que recibe el estado de 2 ascensores a1 y a2. Devuelve "Libre" si entre a1 y a2 hay algún ascensor disponible. En caso contrario, devuelve "Ocupado".

-optimo(x,y,n) es una función que recibe el estado de 2 ascensores a1 y a2, y un numero entre 0 y 9 que representa el piso destino. Esta función implementará una estrategía para decidir cual ascensor debe ser en el encargado de ir hasta el piso destino. Devuelve "a1" si el ascensor óptimo es a1 o "a2" en el caso contrario.

2.4 Dev de botonera

BotoneraDev$\langle X, Y, S, \delta int, \delta ext, \lambda, ta\rangle$

$X = \{0..9\} \times \{0\} \cup \{"Libre", "Ocupado"\} \times \{1\}$
$Y = \{0..9\}$
$S = \{"Libre", "Ocupado"\} \times$ Cola de N $\times \mathrm{R}^+$

El estado S está compuesto por el estado de los ascensores $\times$ peticiones a tratar $\times$ tiempo.

$$\delta ext(s, e, x) = \delta ext((estado, Q, t), e, (n, p))$$
$$\begin{cases} (estado, Q.add(n), 0) & \text{si p=0 } \wedge estado = "Libre" \\ (estado, Q.add(n), \infty) & \text{si p=0 } \wedge estado = "Ocupado" \\ (n, Q, 0) & \text{si p1= } \wedge n = "Libre" \\ (n, Q, \infty) & \text{si p=1 } \wedge n = "Ocupado" \end{cases}$$

$$\delta int(s) = \delta int(estado, Q, t)$$
$$\begin{cases} (estado, Q.pop(), \infty) & \text{si estado="Libre" } \wedge \neg Q.isEmpty() \end{cases}$$

$$\lambda(s) = \lambda(estado, Q, tiempo) =$$
$$\begin{cases} Q.top() & \text{si estado="Libre" } \wedge \neg Q.isEmpty() \end{cases}$$

$ta(s) = ta(estado, Q, t) = t$

$SInicial("Libre", \emptyset, \infty)$

2.5 Dev de generador

GeneradorDev$\langle X, Y, S, \delta int, \delta ext, \lambda, ta \rangle$

$X = \emptyset$
$Y = \{0..9\}$
$S = R^+ \times R^+$

$\delta ext(s, e, x) = \{\}$

$\delta int(s) = \delta int(sem, tiempo) = (rn(sem, 0, 1), (\frac{-1}{\frac{1}{9}} ln(1 - sem))$

$\lambda(s) = \lambda(sem, tiempo)$

$$\begin{cases} 0 & \text{si sem} < \frac{1}{10} \\ 1 & \text{si } \frac{1}{10} \leq \text{sem} < \frac{2}{10} \\ 2 & \text{si } \frac{2}{10} \leq \text{sem} < \frac{3}{10} \\ 3 & \text{si } \frac{3}{10} \leq \text{sem} < \frac{4}{10} \\ 4 & \text{si } \frac{4}{10} \leq \text{sem} < \frac{5}{10} \\ 5 & \text{si } \frac{5}{10} \leq \text{sem} < \frac{6}{10} \\ 6 & \text{si } \frac{6}{10} \leq \text{sem} < \frac{7}{10} \\ 7 & \text{si } \frac{7}{10} \leq \text{sem} < \frac{8}{10} \\ 8 & \text{si } \frac{8}{10} \leq \text{sem} < \frac{9}{10} \\ 9 & \text{si } \frac{9}{10} \leq \text{sem} < \frac{10}{10} \end{cases}$$

$ta(s) = ta(sem, tiempo) = tiempo$

$SInicial = (4, 0)$

3 Devs acoplados

3.1 Dev acoplado para 1 ascensor

N1 $\langle X, Y, D, \{Md\}, EIC, EOC, IC, Select\rangle$

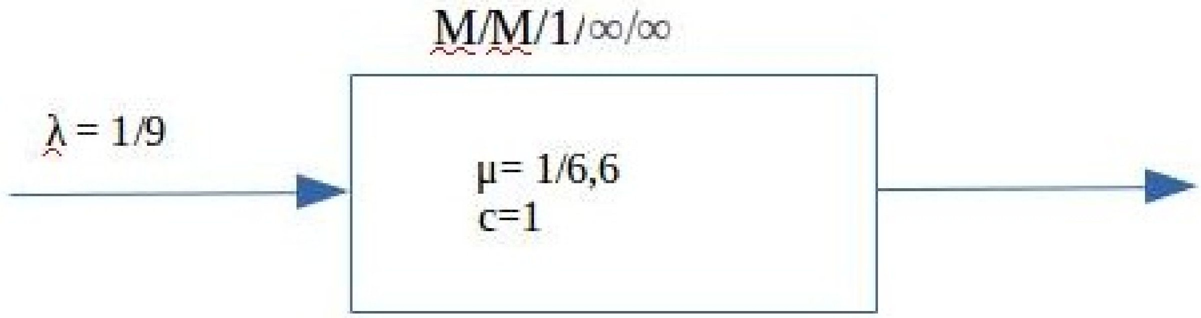
$X = \emptyset$
$Y = \emptyset$
$D = \{Ascensor, Controlador, Botonera, Generador\}$
$\{Md\} = \{Ascensor \rightarrow AscensorDev, Controlador \rightarrow Controlador1Dev, Botonera \rightarrow BotoneraDev,$
$Generador \rightarrow GeneradorDev\}$
$EIC = \emptyset$
$EOC = \emptyset$
$IC = \{[(Generador, 0), (Botonera, 0)], [(Botonera, 0), (Controlador, 0)], [(Controlador, 0), (Ascensor, 0)],$
$[(Controlador, 1), (Botonera, 1)], [(Ascensor, 0), (Controlador, 1)]\}$
$Select = (Ascensor, Controlador, Botonera, Generador)$

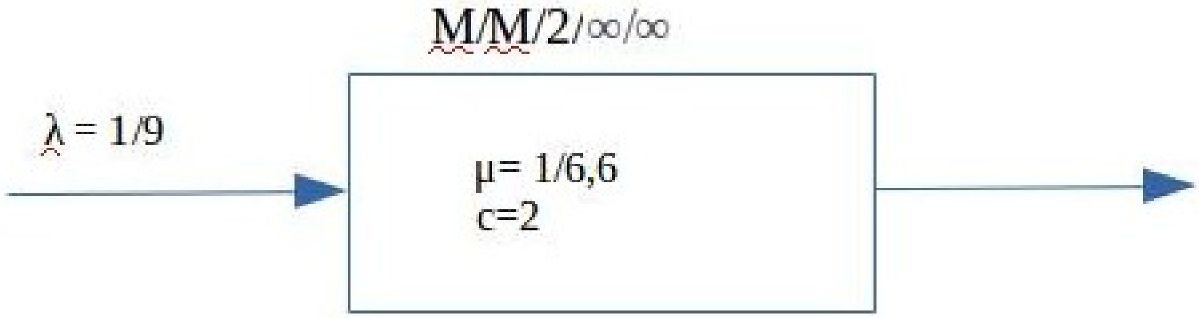
3.2 Dev acoplado para 2 ascensores

N2 $\langle X, Y, D, \{Md\}, EIC, EOC, IC, Select\rangle$

$X = \emptyset$
$Y = \emptyset$
$D = \{Ascensor1, Ascensor2, Controlador, Botonera, Generador\}$
$\{Md\} = \{Ascensor1 \rightarrow\ AscensorDev, Ascensor2 \rightarrow AscensorDev, Controlador \rightarrow Controlador2Dev,$
$Botonera \rightarrow BotoneraDev, Generador \rightarrow GeneradorDev\}$
$EIC = \emptyset$
$EOC = \emptyset$
$IC = \{[(Generador, 0), (Botonera, 0)], [(Botonera, 0), (Controlado, 0)], [(Controlador, 0), (Ascensor1, 0)],$
$[(Controlador, 1), (Ascensor2, 0)], [(Ascensor1, 0), (Controlador, 1)], [(Ascensor2, 0), (Controlador, 2)],$
$[(Controlador, 2), (Botonera, 1)]\}$
$Select = (Ascensor1, Ascensor2, Controlador, Botonera, Generador)$

4 Análisis de cola





Tenemos como dato que el ascensor se mueve a 3,3 pisos (en promedio) para atender a una petición, como cada piso tiene 2 metros entonces el ascensor recorrer 6,6 metros por cada petición a 1 m/s. Es lo mismo decir que el tiempo promedio de servicio es 6,6s.
Con lo que podemos deducir que $\mu = \frac{1}{6{,}6}$.

Para ver si el sistema es estable tengo que analizar si $\lambda < \mu.c$
(1) $\frac{1}{9} < \frac{1}{6{,}6}.1 \rightarrow 0,11 < 0,15$
(2) $\frac{1}{9} < \frac{1}{6{,}6}.2 \rightarrow 0,11 < 0,30$

Luego, los dos sistemas son estables.

4.1 Cálculo de las variables estadísticas

Las variables estadísticas para el sistema de colas con un sólo ascensor son

M/M/c Queue		
lambda	0.111	(arrival rate)
mu	0.152	(service rate)
c	1.000	(number of servers)
rho	0.733	(utilization)
L	2.750	(mean number in system)
w	24.750	(mean time in system)
wQ	18.150	(mean time in queue)
LQ	2.017	(mean number in queue)
P0	0.267	(probability of an empty system)

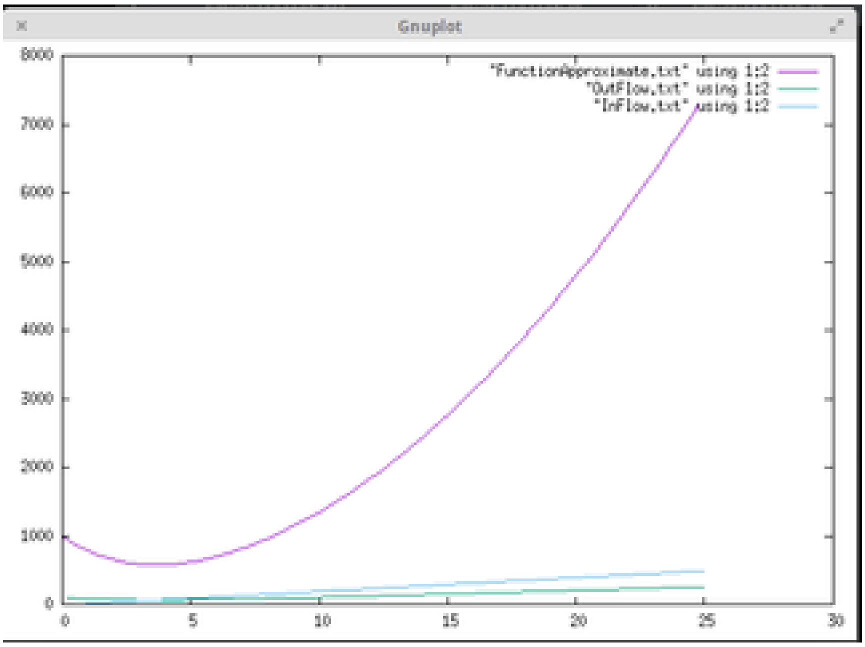
Las variables estadísticas para el sistema de colas con 2 ascensores son

M/M/c Queue		
lambda	0.111	(arrival rate)
mu	0.152	(service rate)
c	2.000	(number of servers)
rho	0.367	(utilization)
L	0.847	(mean number in system)
w	7.625	(mean time in system)
wQ	1.025	(mean time in queue)
LQ	0.114	(mean number in queue)
P0	0.463	(probability of an empty system)

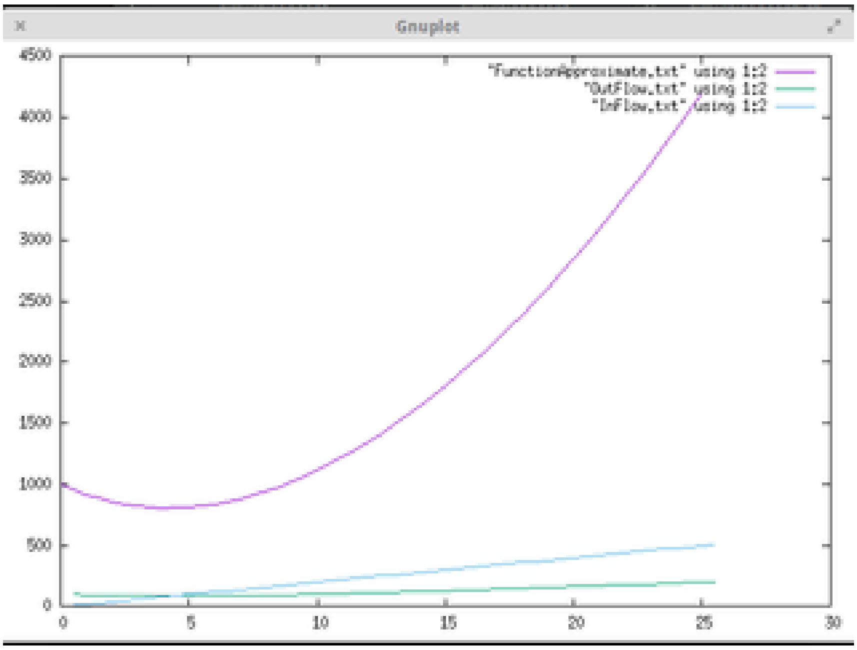
5 Análisis del sistema continuo a implementar

El ejercicio a implementar es el ejercicio 5 de la practica 8 de sistemas continuos. A continuación realizaremos un análisis de los resultados obtenidos mediante el programado, que hemos realizado. El programa recibe como parámetro el tiempo total de simulación, tiempo de avance, y altura inicial del agua del tanque.

Al ejecutar el programa con parametros, tiempo total = 25,0, tiempo de avance = 0,2 y altura inicial del aguad del tanque=1000,0 obtenemos el siguiente grafico.



Ahora si ejecutamos el programa con parametros, tiempo total = 25,0, tiempo de avance = 0,5 y altura inicial del aguad del tanque=1000,0 obtenemos el siguiente grafico.



Analizando los gráficos podemos observar que si tomamos tiempos de avance distintos la función que estamos aproximando, altura del tanque, sufre cambios importantes.

En una primer prueba tomamos un paso de 0,2 y el mínimo valor que toma la función se aproxima a 500. Mientras que si analizamos la segunda prueba con paso 0,5, el punto mínimo que se puede observar que ronda los 800.

Ésto nos permite decir que la variación del intérvalo de aproximación influye fuertemente en la exactitud de la aproximación.