# Proyecto Simulación

## Modelado de un Sistema de Ascensores

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.

Desarrollar tres políticas diferentes para los ascensores, luego realizar un análisis de los tiempo de espera de un ascensor.

1. Los pedidos se alternan si o si entre los 2 ascensores.
2. Se envía siempre a uno disponible, priorizando el elevador 1.
3. Se envía al elevador que según ciertos cálculos llegara más rápido al piso detino.

Ezequiel Depetris - Gaston Massimino

Devs Ascensor $S_{init}$ (1,0,0,∞,false)

$X = \{subir, bajar, parar\} x \{0\}$

//entrada del controlador

$Y = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} x \{0\}$

//salida del piso corriente para el controlador

$S = (IRxIRx\{subiendo, bajando, parado\}xIR)$

//piso corriente, distancia al próximo piso, estado, sigma

$\delta_{ext}(S,e,X)=\delta_{ext}((pc,dps,est,\sigma),e,x=$

(p,2,subiendo,2) si est=parado^X=subir//ascensor parado y llega un subir

(p,2,bajando,2) si est=parado^X=bajar//ascensor parado y llega un subir

(p,0,parado,∞) si est≠parado^X=parar//ascensor en movimiento y llega un parar

(p,dps-V.e,est,2-V.e) si est≠parado^X≠parar//ascensor en movimiento y llega un subir o bajar

$$\delta_{int}(S)=\delta_{int}(pc,dps,est,\sigma)=$$

$$\begin{cases} (pc+1,2,subiendo,2) & \text{si} \quad est=subiendo \\ (pc-1,2,bajando,2) & \text{si} \quad est=bajando \\ (p,2,parado,\infty) & \text{si} \quad est=parar \end{cases}$$

$$Ta(s)=Ta(pc,dps,est,\sigma)=\sigma$$

$$\lambda(s)=\lambda\,(pc,dps,est,\sigma)= pc$$

Devs Controlador $S_{init}$ (1,1,libre,∞,false)

$X = (IR \times \{0,1\})$
//numero de piso por el puerto 0 o 1

$Y = (\{subir, bajar, parar\} x0) \cup (\{libre\} x1)$
//salida para ascensor por puerto 0, salida para panel por puerto 1

$S = (IRxIRxIRx\{0,1\})$
//piso corriente, piso destino, sigma, flag

$\delta_{ext}(S,e,X)=\delta_{ext}((pc,pd,\sigma,output),e,(valor,puerto))=$

(pc,valor,0,0) si puerto=0 //nro de piso que viene del panel

(valor,pd,0,0) si puerto=1^valor=pd //piso que viene del ascensor y es el destino

(valor,pd,∞,0) si puerto=1^valor≠pd //piso que viene del ascensor y no es el destino

$Ta(s)=Ta\ (pc,pd,\sigma,output)=\sigma$

$\lambda(s)=\lambda(pc,pd,\sigma,output)=$

| | | |
|---|---|---|
| $(0,subir)$ | si $pc<pd \wedge output=0$ | //debe seguir subiendo |
| $(0,bajar)$ | si $pc>pd \wedge output=0$ | //debe seguir bajando |
| $(0,parar)$ | si $pc=pd \wedge output=0$ | //debe para porque llego al piso |
| $(1,libre)$ | si $pc=pd \wedge output=1$ | //llego a destino y ahora esta libre |

$$\delta_{int}(S)= \begin{cases} (pc,pd,\infty,0) & \text{si output} = 1 \\ (pc,pd,0,1) & \text{si output} = 0 \end{cases}$$

# Devs Tablero - Prioridad $S_{init}$ ([],libre,libre, $\infty$)

$X = (IR \times 1) \cup (\{libre\} \times \{0,2\})$

//numero de piso por el puerto 1, libre/ocupado de los controladores por los puertos 0,2

$Y = IR \times \{0,1\}$

//numero de piso de salida para el ascensor

$S = [IR] \times \{libre, ocupado\} \times \{libre, ocupado\} \times IR$

//cola de pisos, estado controlador1, estado controlador2, sigma

$\delta_{ext}(S,e,X)=\delta^{ext}((ps,est1,est2,\sigma),e,(valor, puero)=$

(valor ▹ps,est1,est2,0) si (est1=libre v est2=libre)^puerto=1

//entrada por puerto 1 y alguno de los dos controladores libre

(valor ▹ps,est1,est2,∞) si est1=ocupado^est2=ocupado^puerto=1

//entrada por puerto 1 y los dos controladores ocupados

(ps,valor,est2,0) si ps≠[]^puerto=0

//entrada por puerto 0 y hay pisos en la cola

(ps,est1,valor,0) si ps≠[]^puerto=2)

//entrada por puerto 2 y hay pisos en la cola

(ps,valor,est2,∞) si ps=[]^puerto=0

//entrada por puerto 0 y no hay pisos en la cola

(ps,est1,valor,∞) si ps=[]^puerto=2)

//entrada por puerto 2 y no hay pisos en la cola

$$Ta(s)=Ta\ (ps,est1,est2,\sigma)=\sigma$$

$$\lambda(s)=\lambda\ \ (ps,est1,est2,\sigma)= \left.\begin{array}{l} (x,0) \text{ si } est1=libre \\ (x,1) \text{ si } est2=libre \end{array}\right\{ \text{where } [pisos] = xs \triangleright x$$

$$\delta_{int}(S)=$$

$$\begin{array}{ll} (xs,ocupado,est2,\infty) & \text{si } est1 = libre \\ (xs,est1,ocupado,\infty) & \text{si } est2 = libre \end{array}$$

# Devs Tablero - Alternado $S_{init}$([],libre,libre,2,∞)

X =(IR x 1) U ({libre} x {0,2})

//numero de piso por el puerto 1, libre/ocupado de los controladores por los puertos 0,2

Y =IR x {0,1}

//numero de piso de salida para el ascensor

S =[IR] x {libre, ocupado} x {libre, ocupado} x IR

//cola de pisos, estado controlador1, estado controlador2, ultimo controlador, sigma

$\delta_{ext}(S,e,X)=\delta_{ext}$ ((ps,est1,est2,last,σ),e,(valor,puerto)=

(valor ▹ ps,est1,est2,last,0) si ((est1=libre ^ last=2)v(est2=libre ^ last=1))^puerto=1

//entrada del generador y el controlador que tiene el turno esta libre

(valor ▹ ps,est1,est2,last,∞) si (est1=ocupado^last=2)v(est2=ocupado^last=1))^puerto=1

//entrada del generador y el controlador que tiene el turno esta ocupado

(xs,libre,est2,last,0) si puerto=0^last=2 ^ xs≠[]

//entrada del controlador1, es su turno y tengo pisos en la cola

(xs,libre,est2,last,∞) si puerto=0^(last=2 v xs=[])

//entrada del controlador1 pero no es su turno o no hay pisos en la cola

(xs,est1,libre,last,0) si puerto=2^last=1 ^ xs≠[]

//entrada del controlador2, es su turno y tengo pisos en la cola

(xs,est1,libre,last,∞) si (puerto=2^(last=2 v xs=[])

//entrada del controlador2 pero no es su turno o no hay pisos en la cola

$Ta(s)=Ta\,(pisos,est1,est2,last,\sigma)=\sigma$

$\lambda(s)= (x,last-1)$

//desencola un piso donde x es xs▹x = [pisos]
//el valor de last-1 indica el puerto de la salida

$$\delta_{Int} = \begin{cases} (xs,ocupado,est2,1,\infty) & \text{si} \quad last=2 \\ (xs,est1,ocupado,2,\infty) & \text{si} \quad last=1 \end{cases}$$

Devs Tablero - Heurística $S_{init}$ ([],libre,libre,1,1,1,1,0,0,∞)

X =(IR x 1) U ({libre} x {0,2})

//numero de piso por el puerto 1, libre/ocupado de los controladores por los puertos 0,2

Y =IR x {0,1}

//numero de piso de salida para el ascensor

S =[IR]x{libre,ocupado}x{libre,ocupado}xIRxIRxIRxIRxIRxIRxIR

// lista o cola de pisos
// estado del controlador 1
// estado del controlador 2
// piso de origen para controlador 1
// piso de origen para controlador 2
// piso de destino para controlador 1
// piso de destino para controlador 2
// tiempo en el que sale el controlador 1
// tiempo en el que sale el controlador 2
// puerto por el que hay sacar algún valor
// sigma

$\delta_{ext}(S,e,X)=\delta_{ext}((ps,est1,est2,po1,po2,pd1,pd2,t1,t2,output,\sigma),e,(valor,puerto)=$

$(valor \triangleright ps,libre,est2,pd1,po2,pd1,pd2,t,t2,output,0)$

si $puerto=1 \wedge ((func()=1 \wedge est1=libre) \vee (func()=2 \wedge est2=libre))$

//entrada del generador, el controlador1 esta libre y es quien primero atenderá el pedido o
// el controlador2 esta libre y es quien primero atenderá el pedido

$(valor \triangleright ps,libre,est2,pd1,po2,pd1,pd2,t,t2,output,\infty)$

si $puerto=1 \wedge ((func()=1 \wedge est1=ocupado) \vee (func()=2 \wedge est2=ocupado))$

//entrada del generador, el controlador1 esta ocupado y es quien primero atenderá el pedido o el controlador2 esta ocupado y es quien primero atenderá el pedido

$(ps,valor,est2,po1,po2,pd1,pd2,t1,t2,output,0)$ si $puerto=0 \wedge ps\neq[] \wedge func()=1$

//entrada del controlador1, hay pisos en la cola y este mismo controlador es quien primero atenderá el pedido

$(ps,valor,est2,po1,po2,pd1,pd2,t1,t2,output,\infty)$ si $puerto=0 \wedge (ps=[] \vee (ps\neq[] \wedge func()=2))$

//entrada del controlador1, hay pisos en la cola pero no atenderá el pedido primero

$(ps,est1,libre,po1,pd2,pd1,pd2,t1,t,output,0)$ si $puerto=2 \wedge ps\neq[] \wedge func()=2$

//entrada del controlador1, hay pisos en la cola y este mismo controlador es quien primero atenderá el pedido

$(ps,est1,libre,po1,pd2,pd1,pd2,t1,t,output,\infty)$ si $puerto=2 \wedge (ps=[] \vee (ps\neq[] \wedge func()=1))$

//entrada del controlador2, hay pisos en la cola pero no atenderá el pedido primero

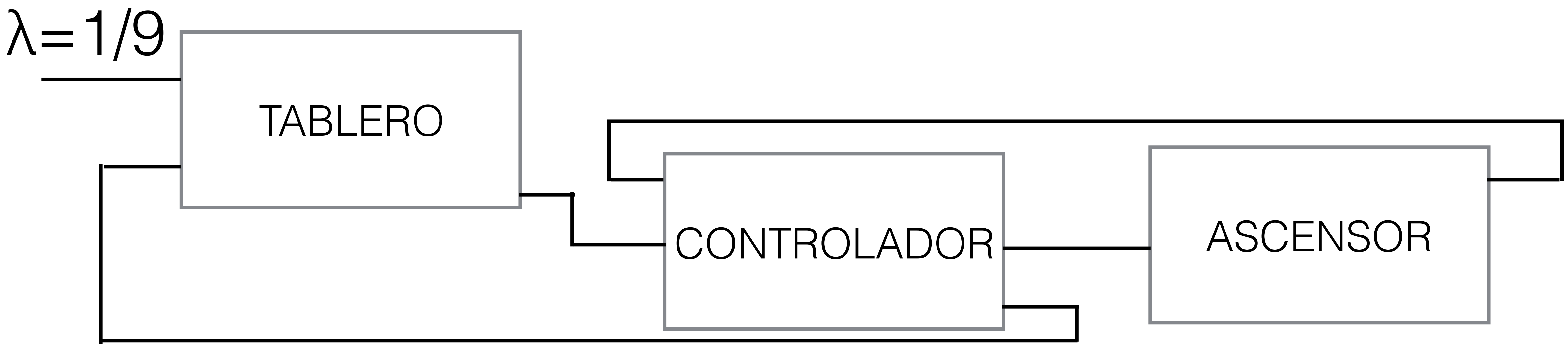
$$\delta_{Int}((ps,est1,est2,po1,po2,pd1,pd2,t1,t2,output,\sigma),e,(valor,puerto)=$$

$$\begin{cases} (ps,est1,est2,po1,po2,pd1,pd2,t1,t2,0,\infty) & \text{si} \quad output = 0 \\ (ps,est1,est2,po1,po2,pd1,pd2,t1,t2,1,\infty) & \text{cc} \end{cases}$$

$$Ta\ (s) = Ta\ (ps,est1,est2,po1,po2,pd1,pd2,t1,t2,output,\sigma) = \sigma$$

$\lambda\ (s) = \lambda\ (ps,est1,est2,po1,po2,pd1,pd2,t1,t2,output,\sigma) = p$ //where $ps \triangleright p$

# Modelo preliminar del problema




ascensor en promedio se mueve 3,3

$$\frac{\lambda}{n * \mu} = \frac{1/9}{1/6.6} < 1 \qquad \text{Cola Estable} \qquad \mu = \frac{1}{6.6}$$

# Análisis de Salida

A continuación se presentan los resultados de las simulaciones discriminando por política de asignación de pedidos a los ascensores que muestran el tiempo medio de espera de un pedido en la cola (tiempo que un pedido espera a ser atendido por alguna ascensor) a partir del método de intervalo de confianza con precisión especifica con un error del 5%.

# Politica Alternante

Se realizaron 7 simulaciones de 15000segundos cada una y se observaron y calcularon los siguientes valores:

Media de las Medias: 16.0711235659862
Varianza: 24.5607904910495

- Aproximando con la distribución normal vemos que el numero mínimo de simulaciones a correr son 37741 ya que la variaza que obtenemos y que sirve como parámetro para aproximar al numero mínimo de simulaciones resulta muy grande.
- Si se quisiera obtener datos mas representativos se deberían hacer mas simulaciones con mucho de mas tiempo ya que la cola parece estar aun en el periodo inestable.

# Politica de Prioridad

Se realizaron 7 simulaciones de 15000segundos cada una y se observaron y calcularon los siguientes valores:

Media de las Medias: 0.813029709470629
Varianza: 0.00438696052113936

- Aproximando con la distribución normal vemos que el numero mínimo de simulaciones a correr son 7 ya que la aproximación arroja como resultado 6.74117901520359.
- Aproximando con la distribución T-student vemos que el primer paso de aproximación casualmente cumple que el valor obtenido es menor o igual que el numero de muestras con el que estamos aproximando por lo que afirmamos que se necesitan 7 simulaciones para obtener el tiempo medio que demora un pedido en ser atendido por cualquier ascensor, con un error del 5%.

# Politica con uso de Heurística

Se realizaron 7 simulaciones de 15000segundos cada una y se observaron y calcularon los siguientes valores:

Media de las Medias: 1.93806445789856
Varianza: 4.73655218220657

- Aproximando con la distribución normal vemos que el numero mínimo de simulaciones a correr son 7279 ya que la variaza que obtenemos y que sirve como parámetro para aproximar al numero mínimo de simulaciones resulta lo suficientemente grande.
- Si se quisiera obtener datos mas representativos se deberían hacer mas simulaciones con mucho de mas tiempo ya que la cola parece estar aun en el periodo inestable.

# Comparación entre las Políticas tomadas

**Política Prioridad:** el tiempo medio de espera de un pedido en la cola se encuentra en el rango de 0.76s a 0.86s con una media de 0.81s.

**Política Alternante:** no se pudo calcular el tiempo medio de espera de un pedido en la cola pero se podría hacer con mas simulaciones y de mayor duración.

**Política Heurística:** no se pudo calcular el tiempo medio de espera de un pedido en la cola pero se podría hacer con mas simulaciones y de mayor duración.