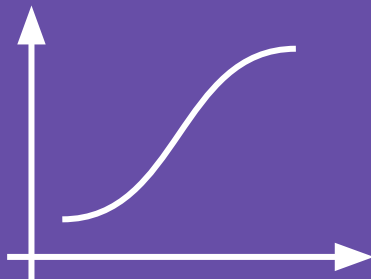


# Modelos y Simulación

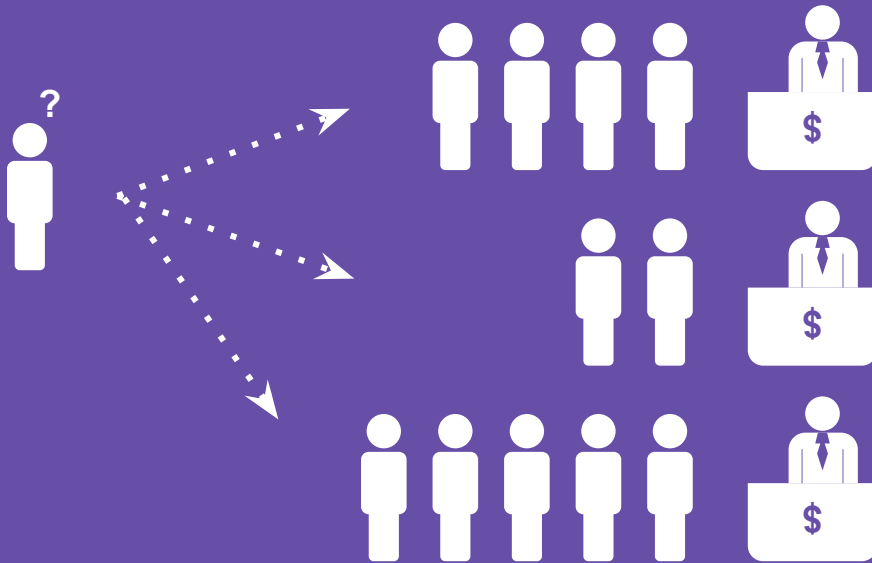
Select  
Funciones



Pablo Montini  
Juan Ignacio Iturriaga  
Franco Lanzillotta

# Modelos y Simulación

Select



Pablo Montini  
Juan Ignacio Iturriaga  
Franco Lanzillota



# Dos cajas con una fila cada una...

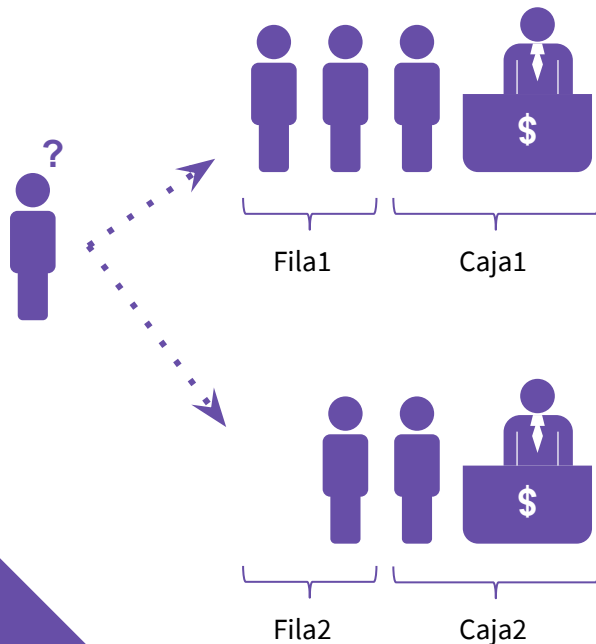
Modelar un sistema con 2 cajas y una fila de espera por cada caja (tipo supermercado), donde los clientes eligen al azar una de las 2 cajas/filas para ser atendidos.



# Dos cajas con una fila cada una...

Modelar un sistema con 2 cajas y una fila de espera por cada caja (tipo supermercado), donde los clientes eligen al azar una de las 2 cajas/filas para ser atendidos.

Situación:

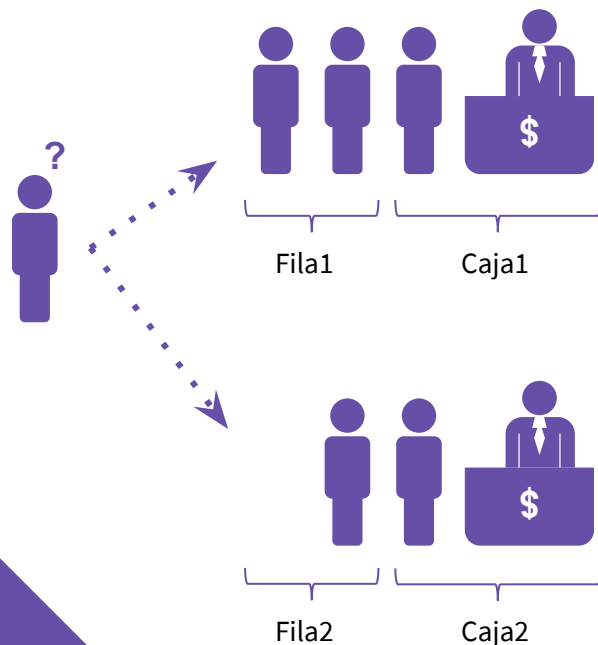




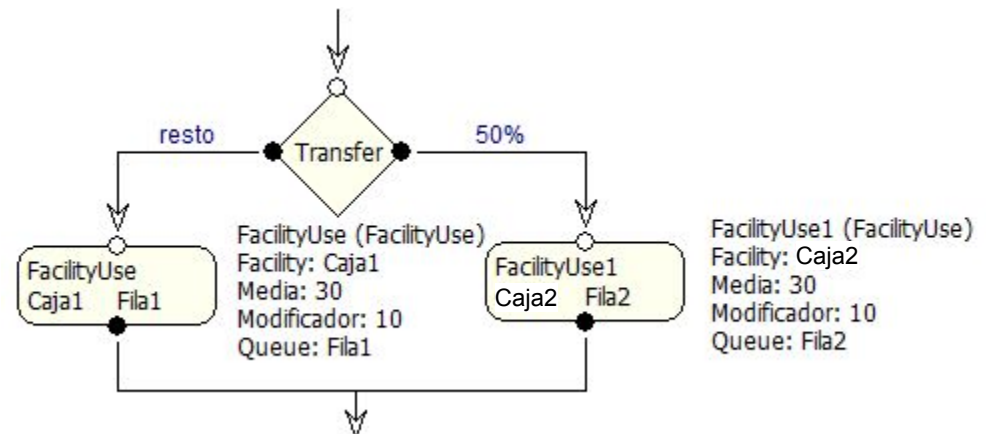
# Dos cajas con una fila cada una...

Modelar un sistema con 2 cajas y una fila de espera por cada caja (tipo supermercado), donde los clientes eligen al azar una de las 2 cajas/filas para ser atendidos.

Situación:



Un posible modelo:

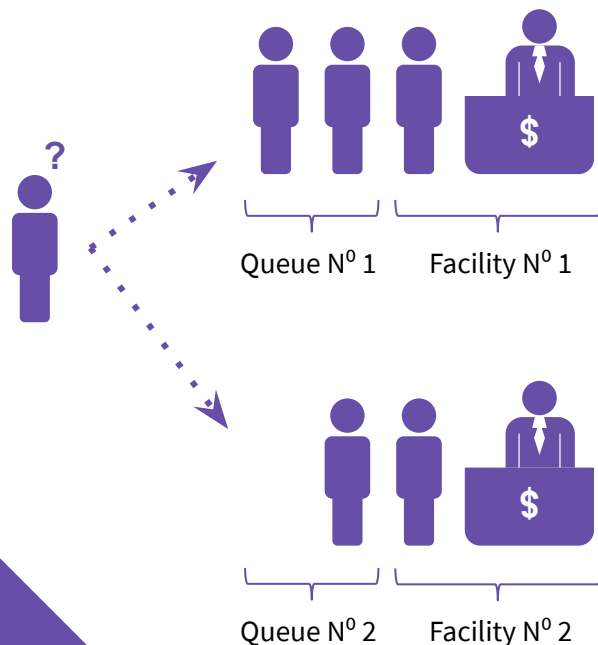




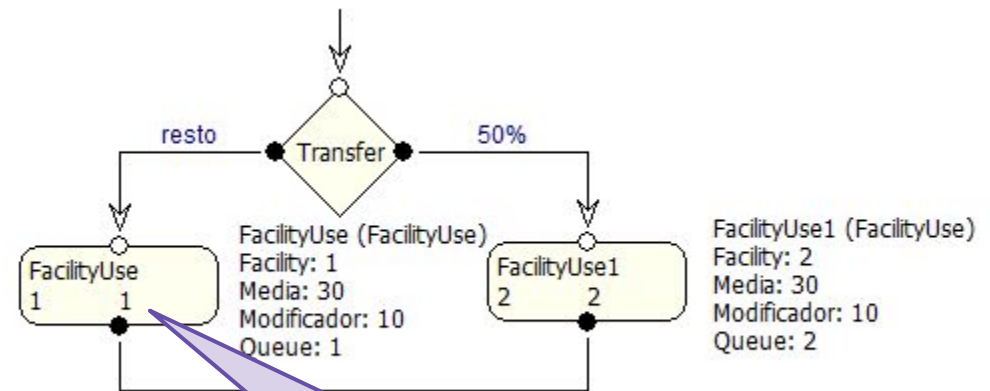
# Dos cajas con una fila cada una...

Modelar un sistema con 2 cajas y una fila de espera por cada caja (tipo supermercado), donde los clientes eligen al azar una de las 2 cajas/filas para ser atendidos.

Situación:



Un posible modelo:



Como con cualquier otra entidad, las Facilities y las Queues también pueden ser identificadas con un número.

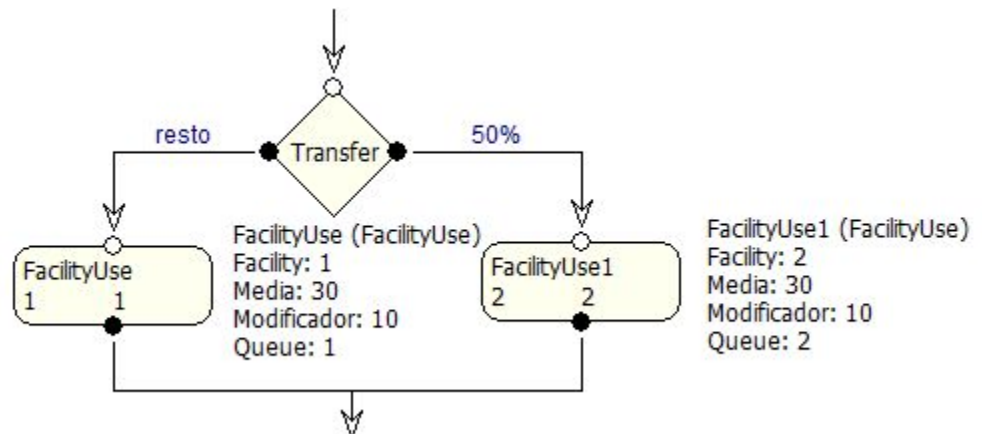
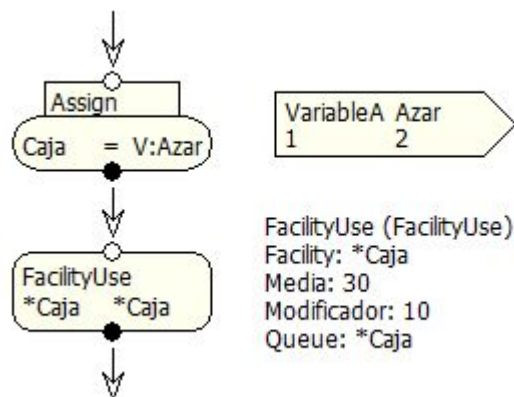


# Dos cajas con una fila cada una...

Modelar un sistema con 2 cajas y una fila de espera por cada caja (tipo supermercado), donde los clientes eligen al azar una de las 2 cajas/filas para ser atendidos.

Solución utilizando parámetros:

Un posible modelo:



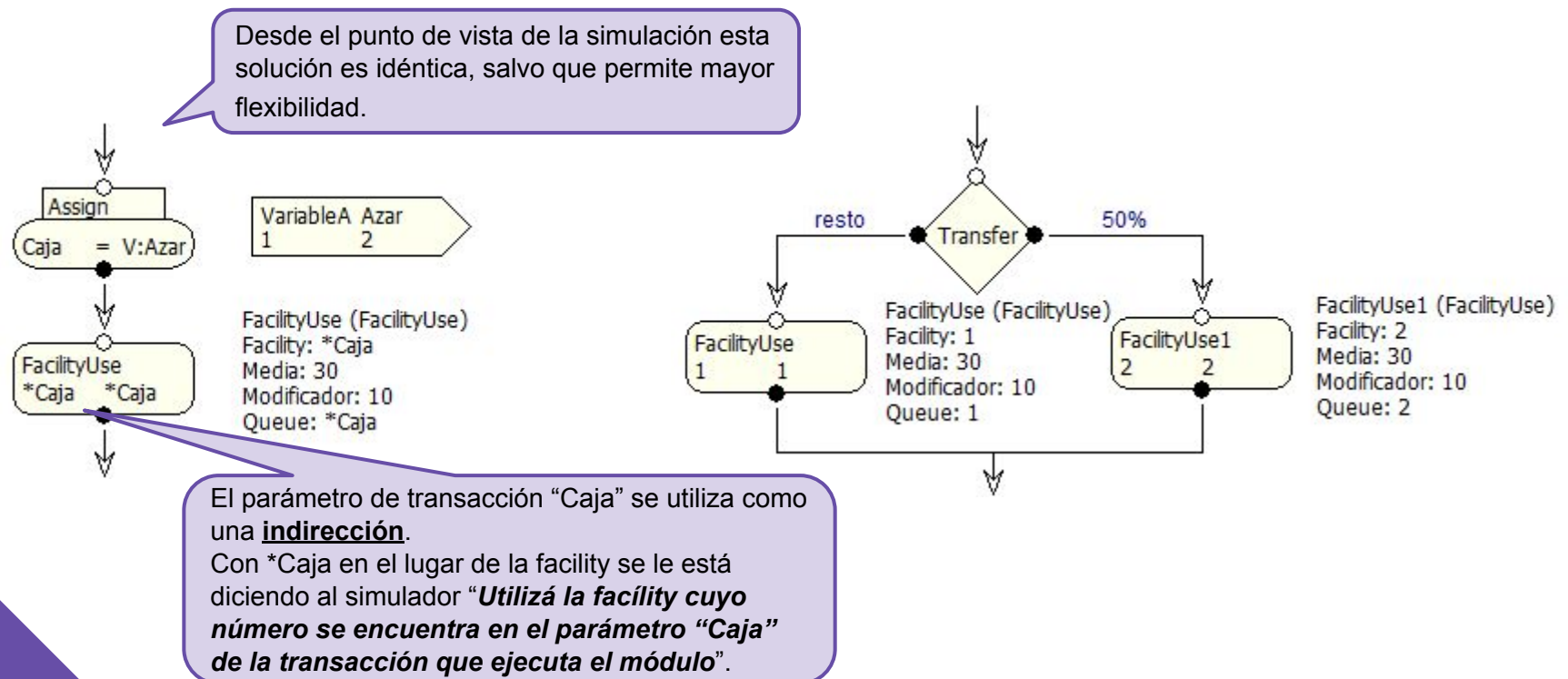


# Dos cajas con una fila cada una...

Modelar un sistema con 2 cajas y una fila de espera por cada caja (tipo supermercado), donde los clientes eligen al azar una de las 2 cajas/filas para ser atendidos.

Solución utilizando parámetros:

Un posible modelo:







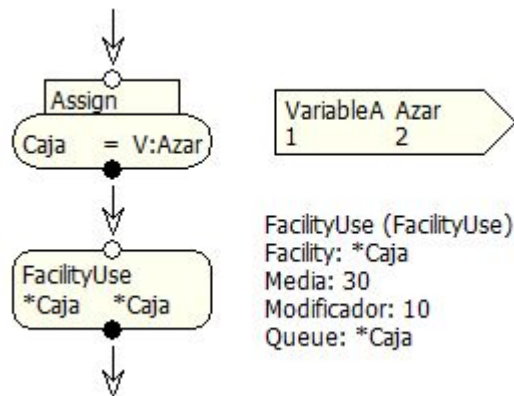
# Dos cajas con una fila cada una...

Modelar un sistema con 2 cajas y una fila de espera por cada caja (tipo supermercado), donde los clientes eligen al azar una de las 2 cajas/filas para ser atendidos.

Solución utilizando parámetros:

Ventajas:

- Se puede cambiar dinámicamente la cantidad de cajas utilizadas
  - Si se cambia el valor máximo de la variable a 3, automáticamente hay 3 cajas habilitadas.



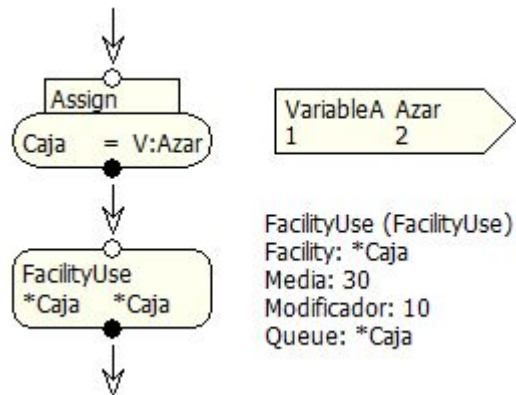


# Dos cajas con una fila cada una...

Modelar un sistema con 2 cajas y una fila de espera por cada caja (tipo supermercado), donde los clientes eligen al azar una de las 2 cajas/filas para ser atendidos.

Solución utilizando parámetros:

Ventajas:



- Se puede cambiar dinámicamente la cantidad de cajas utilizadas
  - Si se cambia el valor máximo de la variable a 3, automáticamente hay 3 cajas habilitadas.
- Se puede utilizar una función para la distribución de elección de cajas.
  - Si el 60% elige la caja 1, el 15% elige la caja 2 y el 25% la caja 3.

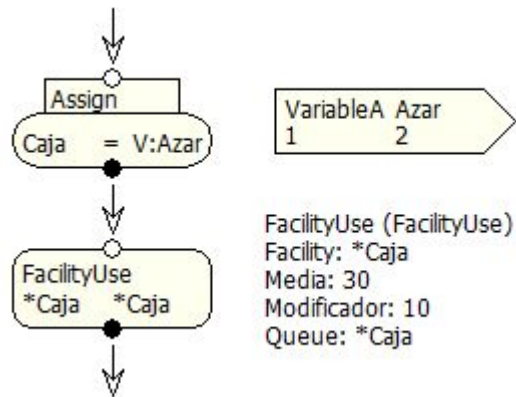


# Dos cajas con una fila cada una...

Modelar un sistema con 2 cajas y una fila de espera por cada caja (tipo supermercado), donde los clientes eligen al azar una de las 2 cajas/filas para ser atendidos.

Solución utilizando parámetros:

Ventajas:



- Se puede cambiar dinámicamente la cantidad de cajas utilizadas
  - Si se cambia el valor máximo de la variable a 3, automáticamente hay 3 cajas habilitadas.
- Se puede utilizar una función para la distribución de elección de cajas.
  - Si el 60% elige la caja 1, el 15% elige la caja 2 y el 25% la caja 3.
- Se podría utilizar otro criterio para la elección de la caja.
  - Por ejemplo: **elegir la caja con menor cantidad de personas en la fila de espera.**

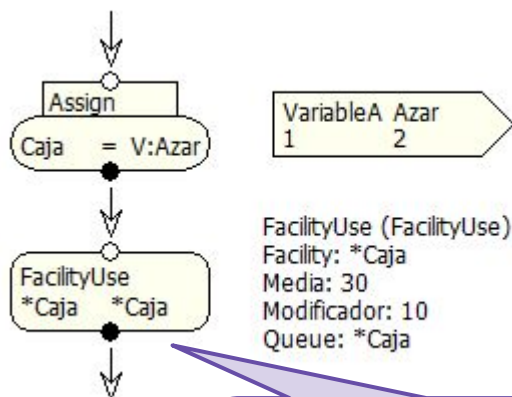


# Dos cajas con una fila cada una...

Modelar un sistema con 2 cajas y una fila de espera por cada caja (tipo supermercado), donde los clientes eligen al azar una de las 2 cajas/filas para ser atendidos.

Solución utilizando parámetros:

Ventajas:



No importa cómo obtuvo la transacción el valor del parámetro "Caja", lo que importa es que tenga un valor **válido** de número de caja (y su respectiva fila) cuando llegue al módulo FacilityUse

- Se puede cambiar dinámicamente la cantidad de cajas utilizadas
  - Si se cambia el valor máximo de la variable a 3, automáticamente hay 3 cajas habilitadas.
- Se puede utilizar una función para la distribución de elección de cajas.
  - Si el 60% elige la caja 1, el 15% elige la caja 2 y el 25% la caja 3.
- Se podría utilizar otro criterio para la elección de la caja.
  - Por ejemplo: **elegir la caja con menor cantidad de personas en la fila de espera.**



# Select

- **Select**

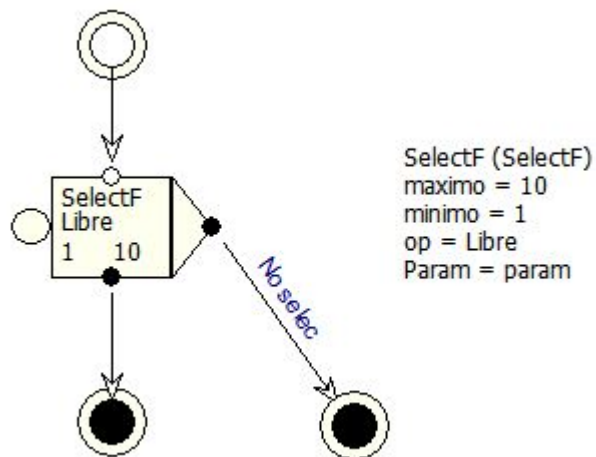
- Es una Instrucción de GPSS
- Permite seleccionar una entidad (referenciada con un número) en base a un rango
  - ¿Qué entidades?
    - F (Facilities)
    - S (Storages)
    - Q (Queues)
    - X (Savevalues)
    - Etc...
- **Devuelve el número de la entidad** seleccionada en **un parámetro de la transacción** que ejecuta el select.
- La entidad seleccionada debe cumplir una condición.

- **Ejemplos:**

- ¿Cual es el Número de Queue que tiene “menos personas”?
- ¿Cuál es el Número de Facility que está “libre”?
- ¿Cuál es el Número de Storage con “mayor espacio libre”?

# SelectF

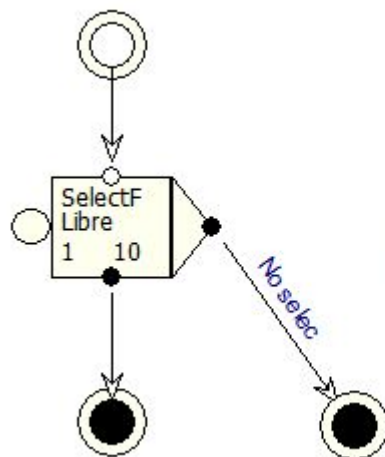
- Selecciona entre las facilities 1 a 10, cual está libre



# SelectF



- Selecciona entre las facilities 1 a 10, cual está libre

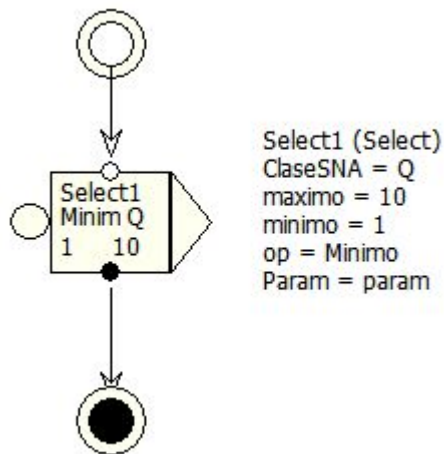


SelectF (SelectF)  
maximo = 10  
minimo = 1  
op = Libre  
Param = param

Propiedades	
Clave	Valor
Clase	SelectF
Nombre	SelectF
Etiqueta	SelectF
maximo	10
minimo	1
op	Libre
Param	Disponible Interrumpida Libre Nodisponible NoInterrumpida Usada

# Select (módulo)

- Selecciona entre las colas 1 a 10, cual es más corta

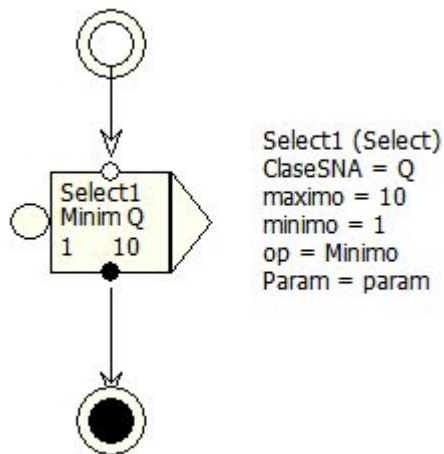






# Select (módulo)

- Selecciona entre las colas 1 a 10, cual es más corta



Propiedades	
Clave	Valor
Clase	Select
Nombre	Select1
Etiqueta	Select1
ClaseSNA	Q
maximo	10
minimo	1
op	Minimo
Param	Maximo Minimo



# Combinación de Selects

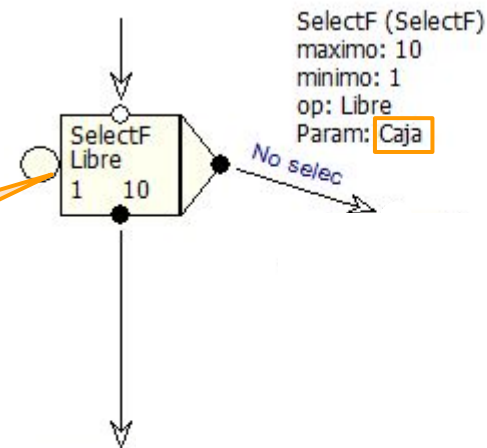
- Sean 10 cajas con sus 10 colas
- Elegir la caja vacía y si no hay ninguna, la caja que tenga cola mínima



# Combinación de Selects

- Sean 10 cajas con sus 10 colas
- Elegir la caja vacía y si no hay ninguna, la caja que tenga cola mínima

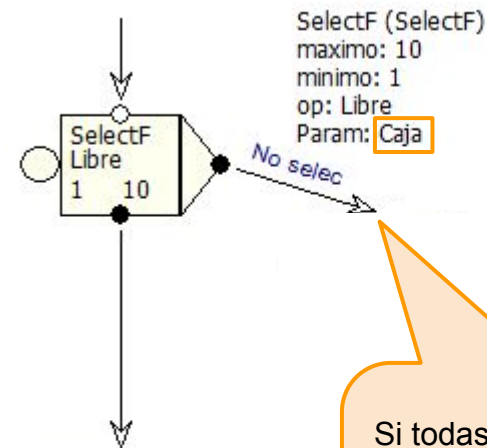
Primero intenta elegir qué facility de la 1 a la 10 está libre.





# Combinación de Selects

- Sean 10 cajas con sus 10 colas
- Elegir la caja vacía y si no hay ninguna, la caja que tenga cola mínima

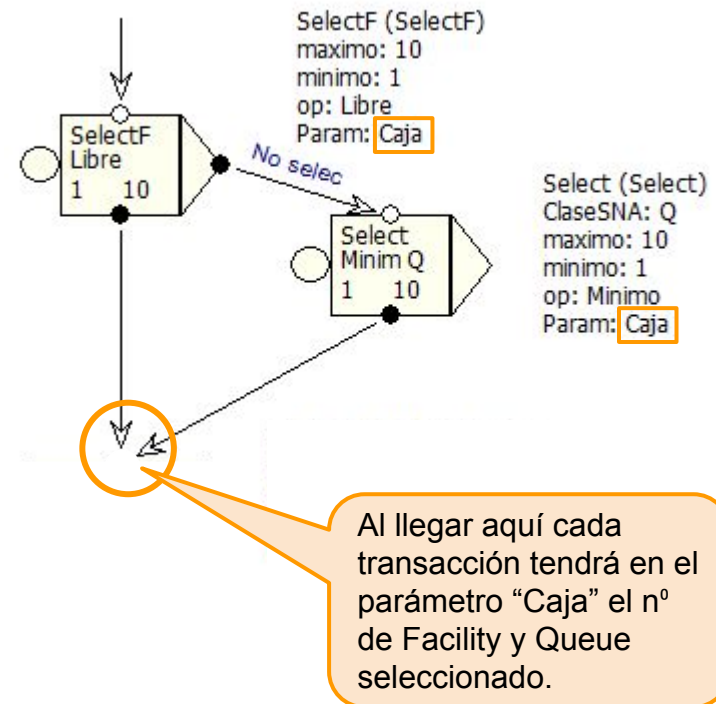


Si todas las facilities están en uso, la transacción saldrá por la salida "No selec". Entonces tendrá que buscar la fila más corta.



# Combinación de Selects

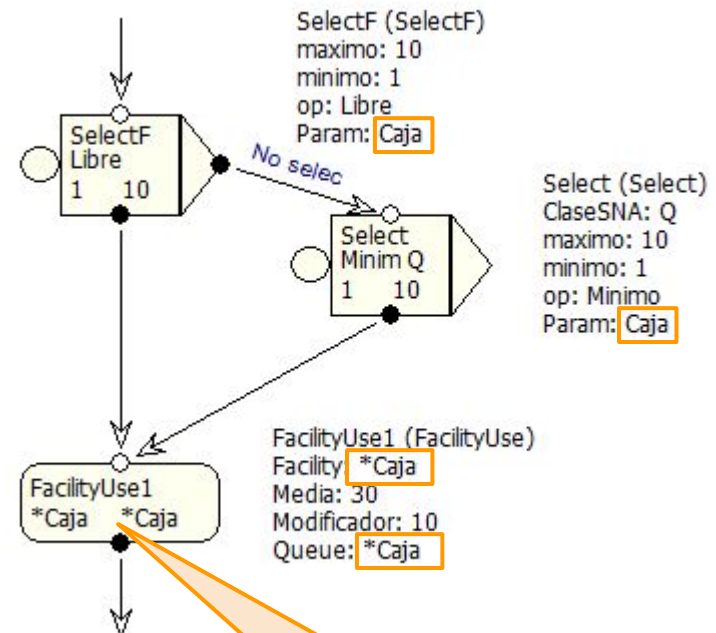
- Sean 10 cajas con sus 10 colas
- Elegir la caja vacía y si no hay ninguna, la caja que tenga cola mínima





# Combinación de Selects

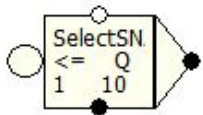
- Sean 10 cajas con sus 10 colas
- Elegir la caja vacía y si no hay ninguna, la caja que tenga cola mínima



Se utiliza el parámetro como **para identificar** ambas entidades. Esto se conoce como **indirección**.



# SelectSNA - SelectS - SelectL



SelectSNA (SelectSNA)  
ClaseSNA = Q  
maximo = 10  
minimo = 1  
op = <=  
Param = param  
valorComp = \*1

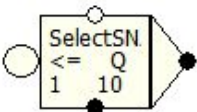
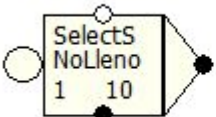
Selecciona entre rango de cualquier SNA

**ClaseSNA:** cualquier SNA de la cartilla

Aplica el **OP**: <, <=, >, >=, =, != para comparar  
contra el valor de referencia indicado en  
**ValorComp**



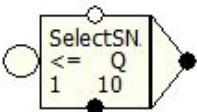
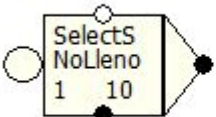
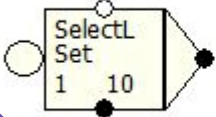
# SelectSNA - SelectS - SelectL

 <p>SelectSNA (SelectSNA) ClaseSNA = Q maximo = 10 minimo = 1 op = &lt;= Param = param valorComp = *1</p>	<p>Selecciona entre rango de cualquier SNA <b>ClaseSNA:</b> cualquier SNA de la cartilla Aplica el <b>OP:</b> &lt;, &lt;=, &gt;, &gt;=, =, != para comparar contra el valor de referencia indicado en <b>ValorComp</b></p>
 <p>SelectS (SelectS) maximo = 10 minimo = 1 op = NoLleno Param = param</p>	<p>Selecciona entre rango de <b>STORAGES</b> <b>OP:</b> F, NF, E, NE, SV, SNV</p>





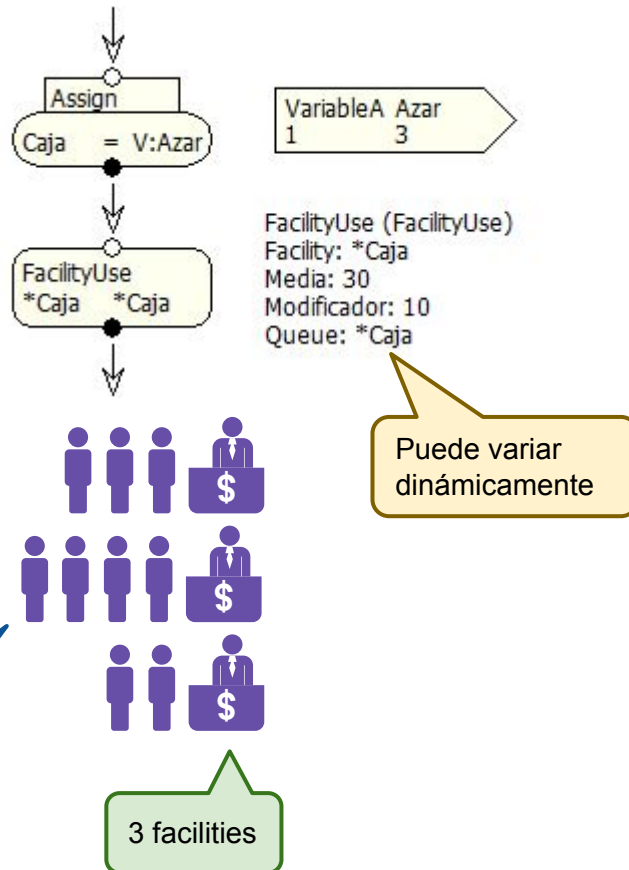
# SelectSNA - SelectS - SelectL

 <p>SelectSNA (SelectSNA) ClaseSNA = Q maximo = 10 minimo = 1 op = &lt;= Param = param valorComp = *1</p>	<p>Selecciona entre rango de cualquier SNA <b>ClaseSNA:</b> cualquier SNA de la cartilla Aplica el <b>OP:</b> &lt;, &lt;=, &gt;, &gt;=, =, != para comparar contra el valor de referencia indicado en <b>ValorComp</b></p>
 <p>SelectS (SelectS) maximo = 10 minimo = 1 op = NoLleno Param = param</p>	<p>Selecciona entre rango de <b>STORAGES</b> <b>OP:</b> F, NF, E, NE, SV, SNV</p>
 <p>SelectL (SelectL) maximo = 10 minimo = 1 op = Set Param = param</p>	<p>Selecciona entre rango de <b>LLAVES LOGICAS</b> <b>OP:</b> Set o Reset</p>

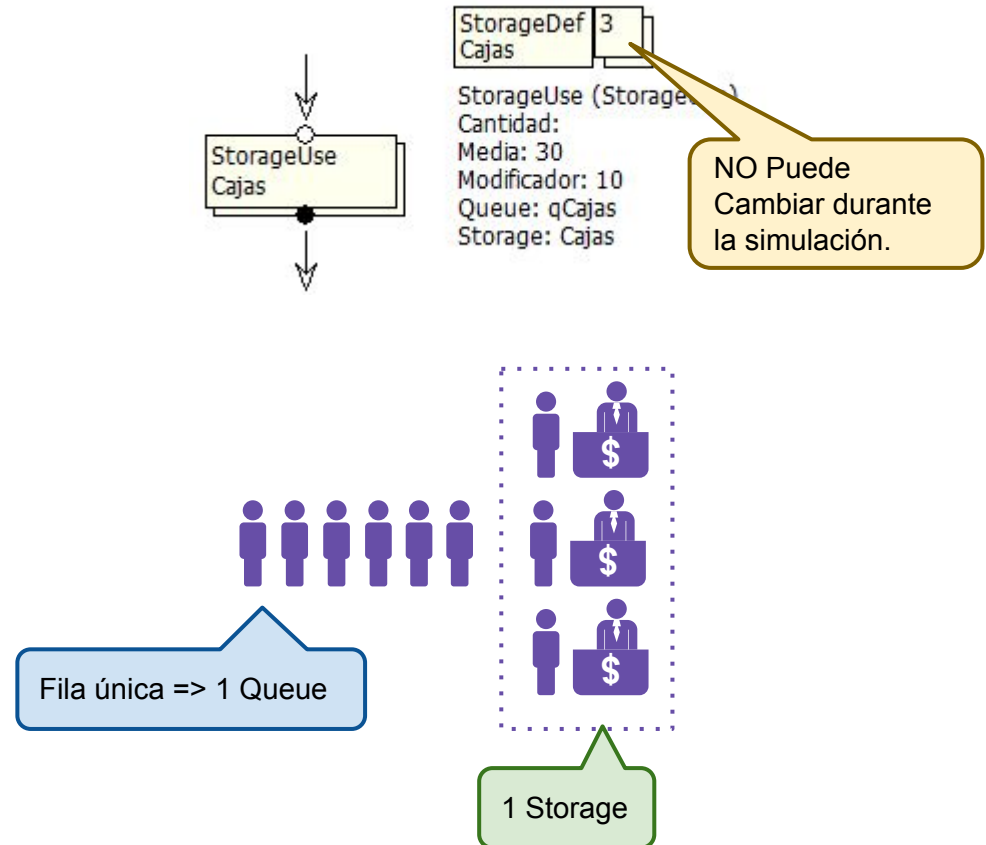
# Facilities vs Storage



## Facilities

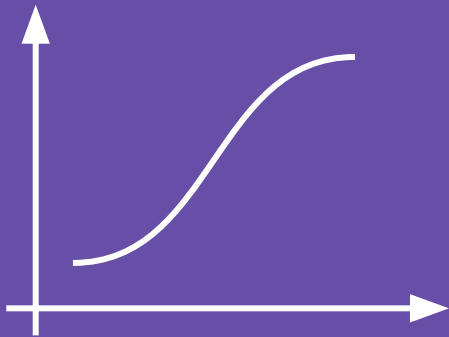


## Storage



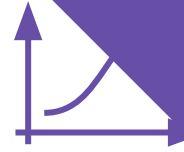
# Modelos y Simulación

Funciones



Pablo Montini  
Juan Ignacio Iturriaga  
Franco Lanzillota

# Funciones



- **Function**

- Es una entidad **global**
- Cada vez que se evalúa una función devuelve un valor
- Puede ser:
  - **Determinística:** al mismo valor de argumento de entrada, siempre retorna el mismo valor de salida.
  - **Estocástica:** al mismo valor de argumento de entrada puede retorna diferente valor de salida.
    - Ejemplo:
      - Cuando la variable de entrada es el SNA  $RN_i$
      - Cuando la función devuelve como resultado otro SNA estocástico

# Funciones



- **Function**

- Es una entidad **global**
- Cada vez que se evalúa una función devuelve un valor
- Puede ser:
  - **Determinística:** al mismo valor de argumento de entrada, siempre retorna el mismo valor de salida.
  - **Estocástica:** al mismo valor de argumento de entrada puede retorna diferente valor de salida.
    - Ejemplo:
      - Cuando la variable de entrada es el SNA  $RN_i$
      - Cuando la función devuelve como resultado otro SNA estocástico

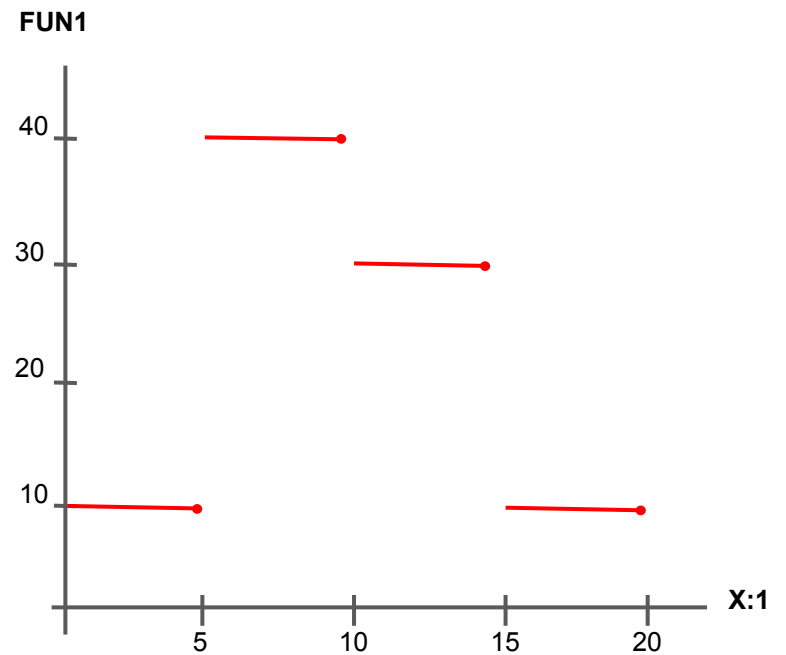
- **Tipos**

- Discreta (escalón)
- Continua
- Evaluación (SNA)

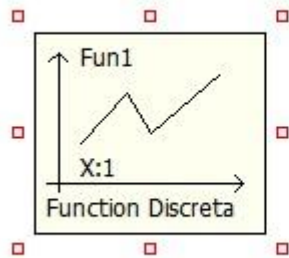
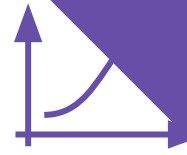
# Funciones Discretas



$$\text{FUN1} = f(\text{X:1})$$



# Funciones Discretas



Function (Function)

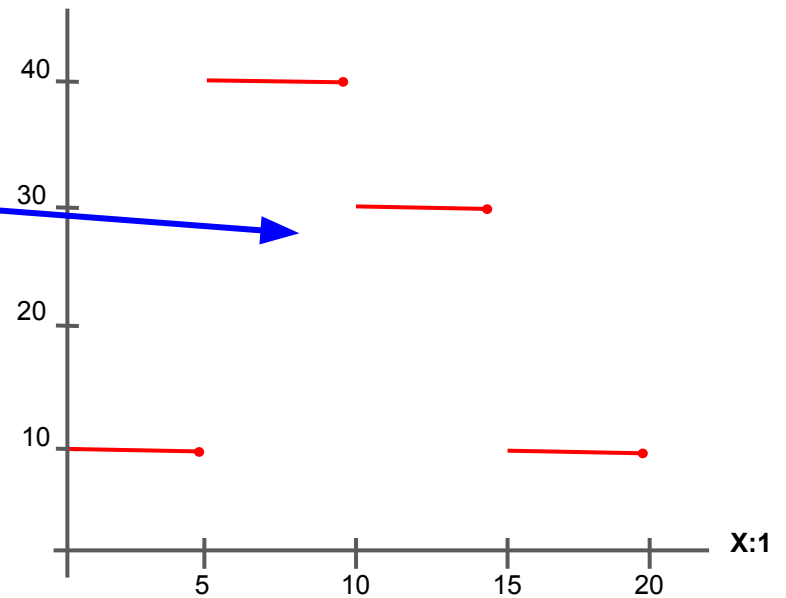
Atributo = X:1

NombreFunc = Fun1

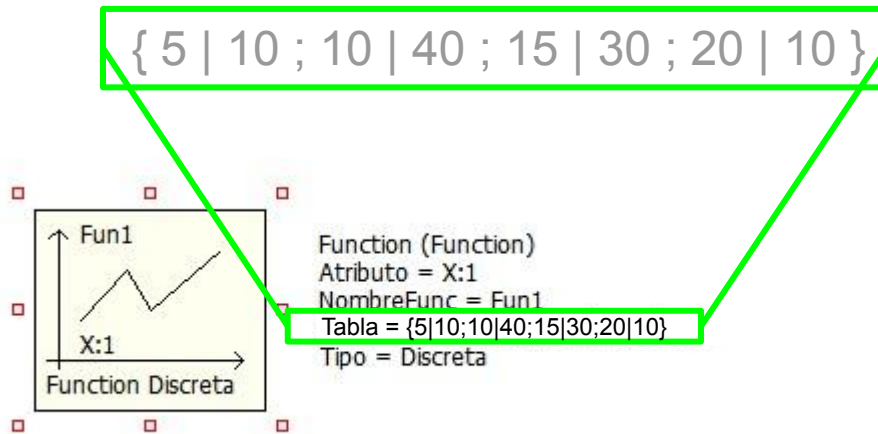
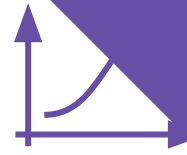
Tipo = Discreta

$FUN1 = f(X:1)$

**FUN1**

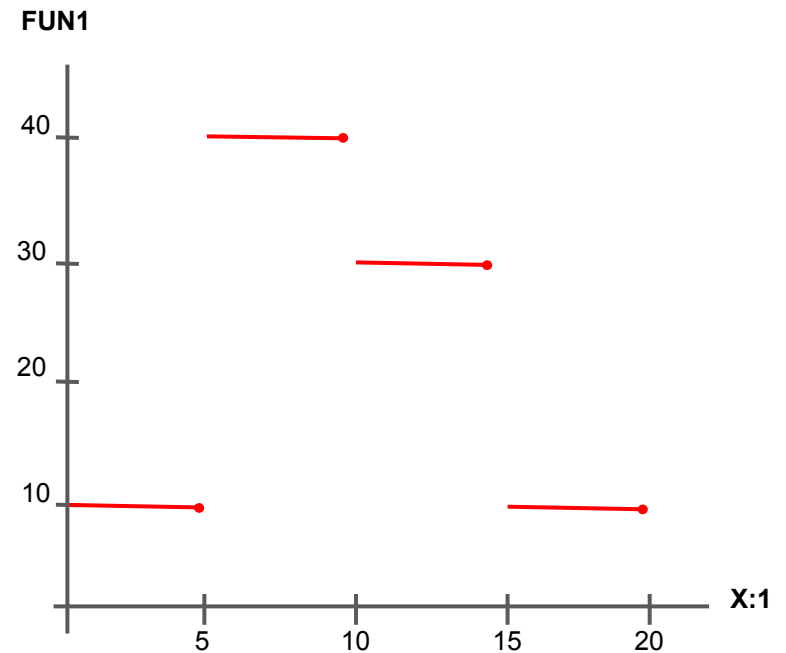


# Funciones Discretas



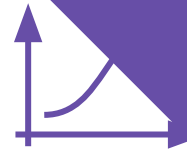
Propiedades	
Clave	Valor
Clase	Function
Nombre	Function
Etiqueta	Function Discreta
Atributo	X:1
NombreFunc	Fun1
Tabla	{5 10;10 40;15 30;20 10}
Tipo	Discreta

$$\text{FUN1} = f(\text{X:1})$$

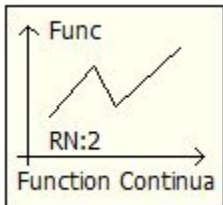




# Funciones Continuas



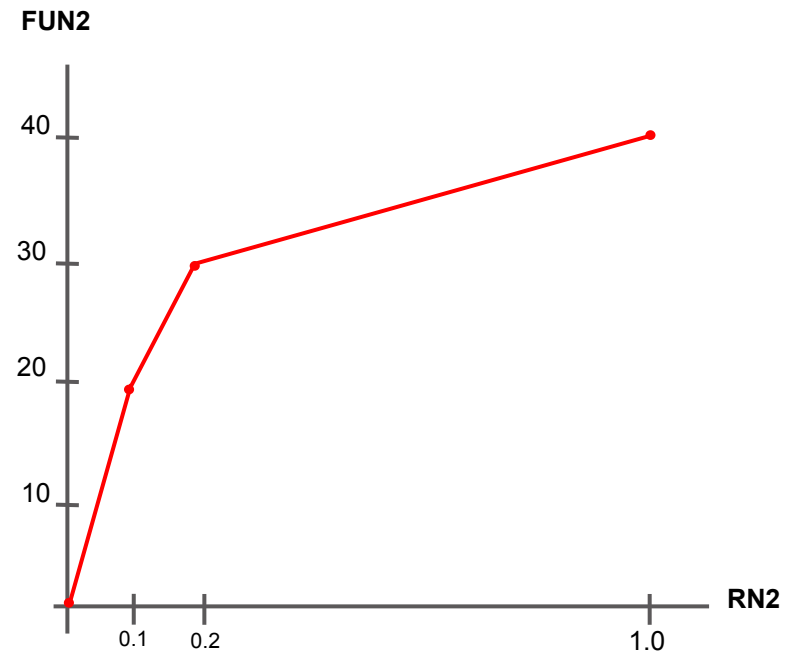
$$\text{FUN2} = f(\text{RN2})$$



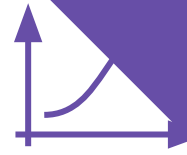
Function (Function)  
Atributo: RN:2  
NombreFunc: Func

Tipo: Continua

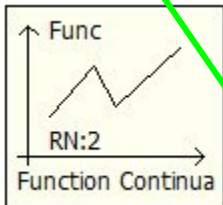
Propiedades	
Clave	Valor
Clase	Function
Nombre	Function
Etiqueta	Function Discreta
Atributo	RN:2
NombreFunc	Fun2
Tabla	{0 0;0.1 20;0.2 30;1 40}
Tipo	Continua



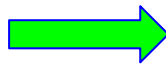
# Funciones Continuas



$\{ 0 \mid 0 ; 0.1 \mid 20 ; 0.2 \mid 30 ; 1 \mid 40 \}$

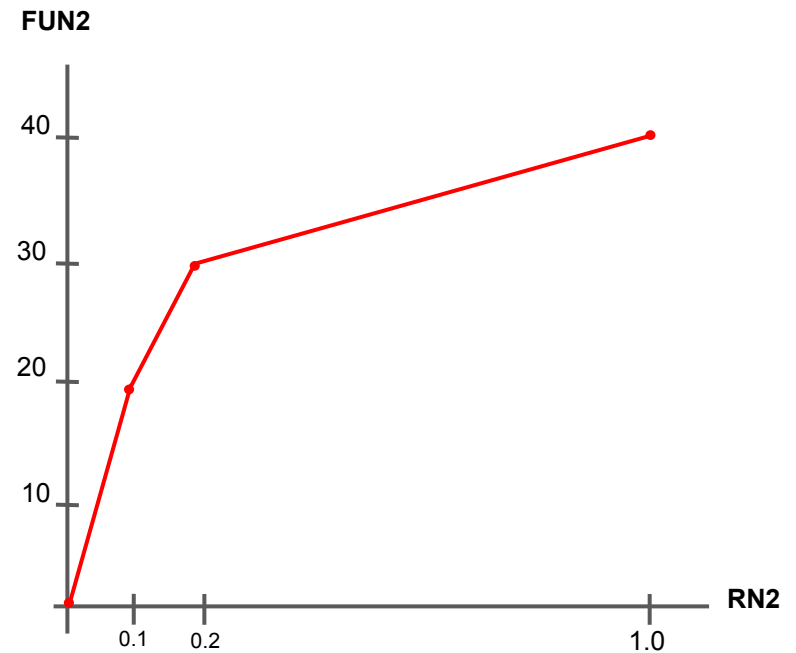


Function (Function)  
 Atributo: RN:2  
 NombreFunc: Func  
 Tabla:  $\{0 \mid 0 ; 0.1 \mid 20 ; 0.2 \mid 30 ; 1 \mid 40\}$   
 Tipo: Continua

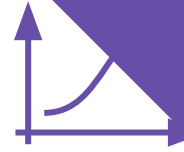


Propiedades	
Clave	Valor
Clase	Function
Nombre	Function
Etiqueta	Function Discreta
Atributo	RN:2
NombreFunc	Fun2
Tabla	$\{0 \mid 0 ; 0.1 \mid 20 ; 0.2 \mid 30 ; 1 \mid 40\}$
Tipo	Continua

$$\text{FUN2} = f(\text{RN2})$$



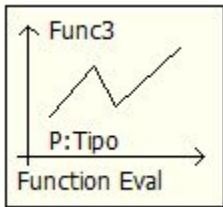
# Funciones Evaluación



$\text{FUN3} = f(\text{P:TIPO})$

- o **si**  $\text{P:TIPO} \leq 1$  **entonces**  
     $\text{FUN3} = \text{S:LOCAL}$
- o **si**  $\text{P:TIPO} == 2$  **entonces**  
     $\text{FUN3} = \text{Q:CAJAS}$
- o **si**  $\text{P:TIPO} > 3$  **entonces**  
     $\text{FUN3} = \text{FR:INFO}$

# Funciones Evaluación



Function (Function)  
Atributo: P:Tipo  
NombreFunc: Func3

Tipo: Eval

Propiedades	
Clave	Valor
Clase	Function
Nombre	Function
Etiqueta	Function Eval
Atributo	P:Tipo
NombreFunc	Func3
Tabla	{1 S:Local;2 Q:Caja
Tipo	Eval

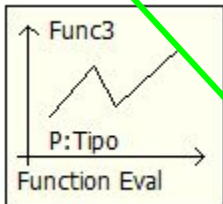
$FUN3 = f(P:TIP0)$

- o **si**  $P:TIP0 \leq 1$  **entonces**  
 $FUN3 = S:LOCAL$
- o **si**  $P:TIP0 == 2$  **entonces**  
 $FUN3 = Q:CAJAS$
- o **si**  $P:TIP0 \geq 3$  **entonces**  
 $FUN3 = FR:INFO$

# Funciones Evaluación



{ 1 | S:LOCAL ; 2 | Q:CAJAS ; 3 | FR:INFO }



Function (Function)  
Atributo: P:Tipo  
NombreFunc: Func3  
Tabla: {1|S:Local;2|Q:Cajas;3|FR:Info}  
Tipo: Eval

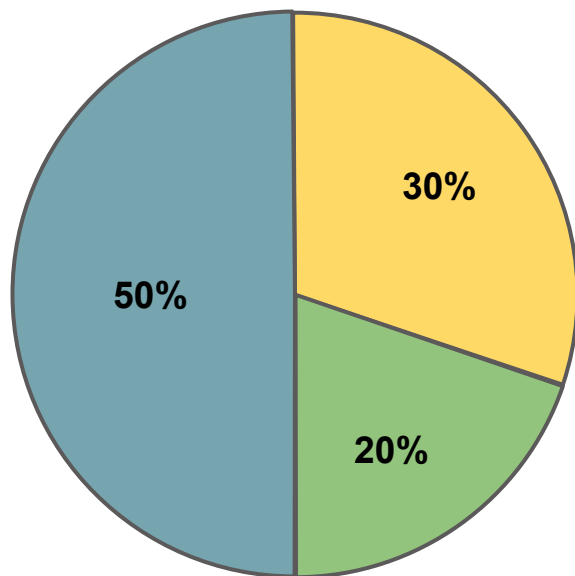
Propiedades	
Clave	Valor
Clase	Function
Nombre	Function
Etiqueta	Function Eval
Atributo	P:Tipo
NombreFunc	Func3
Tabla	{1 S:Local;2 Q:Caje
Tipo	Eval

$$\text{FUN3} = f(\text{P:TIP0})$$

- o si  $\text{P:TIP0} \leq 1$  entonces  
 $\text{FUN3} = \text{S:LOCAL}$
- o si  $\text{P:TIP0} == 2$  entonces  
 $\text{FUN3} = \text{Q:CAJAS}$
- o si  $\text{P:TIP0} \geq 3$  entonces  
 $\text{FUN3} = \text{FR:INFO}$

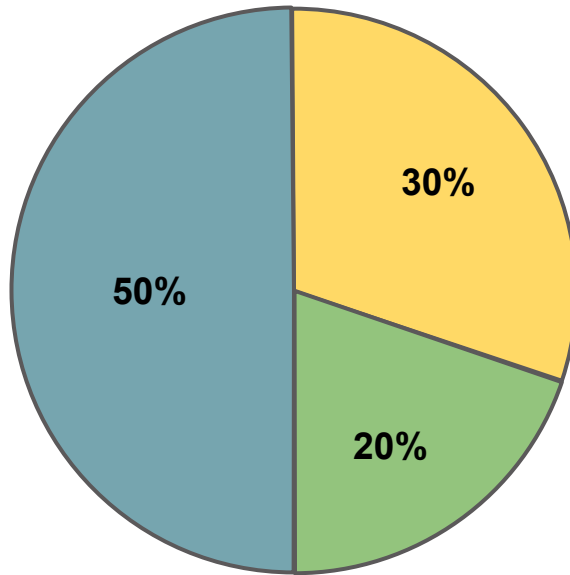
# Función Distribución

- Tipo 1: 30%
- Tipo 2: 20%
- Tipo 3: 50%

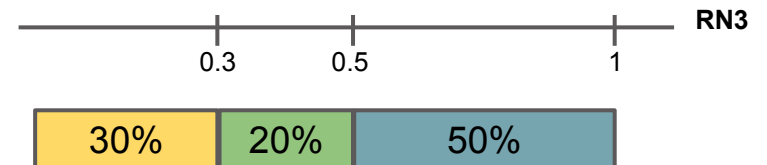


# Función Distribución

- Tipo 1: 30%
- Tipo 2: 20%
- Tipo 3: 50%

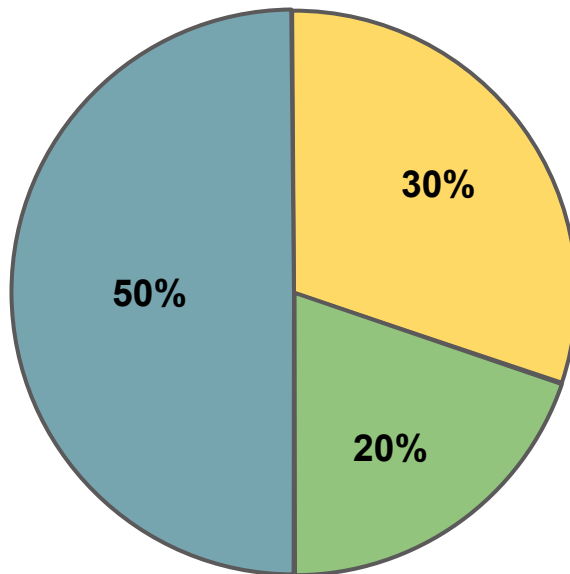


TipoCliente =  $f(RN3)$

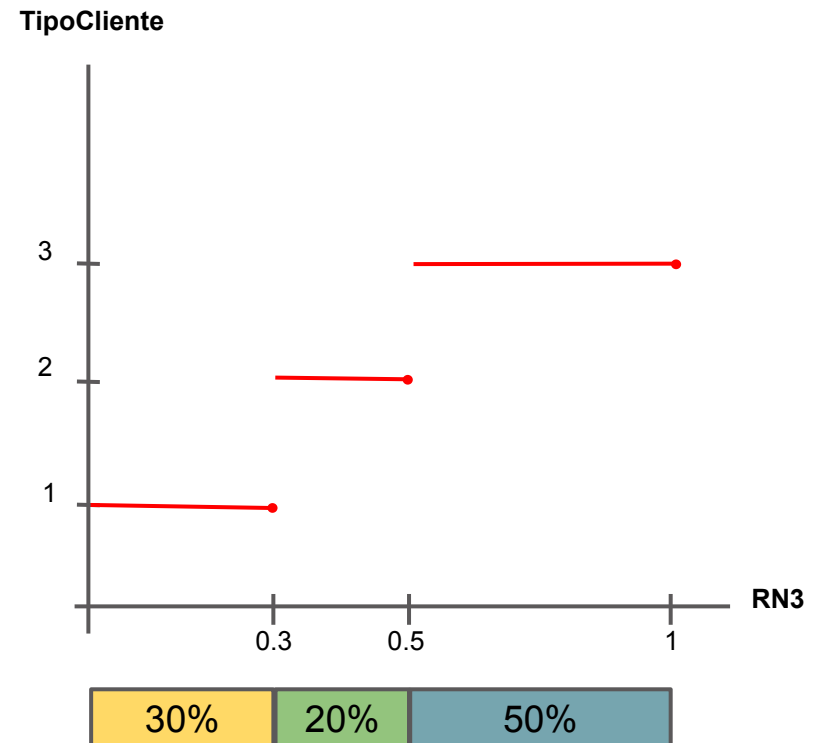


# Función Distribución

- Tipo 1: 30%
- Tipo 2: 20%
- Tipo 3: 50%

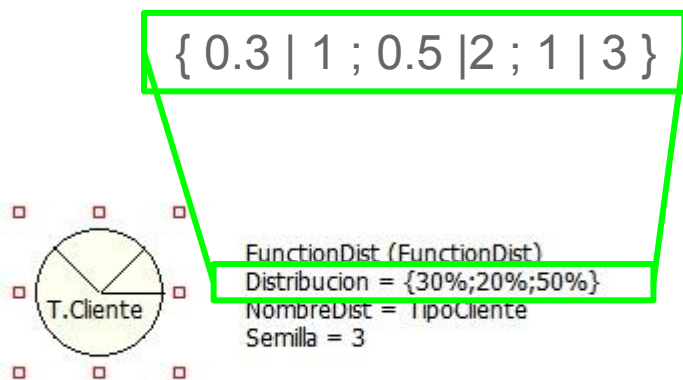


$$\text{TipoCliente} = f(\text{RN3})$$





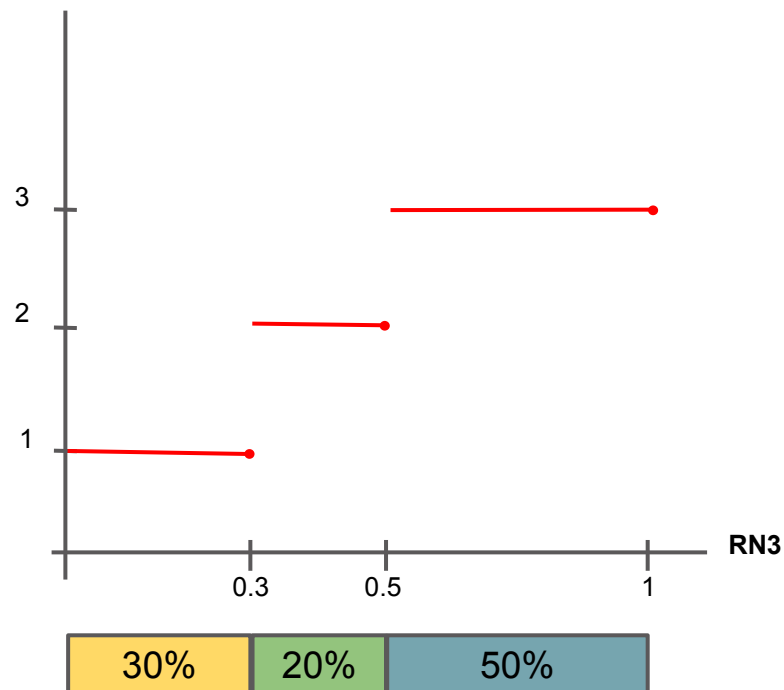
# Función Distribución



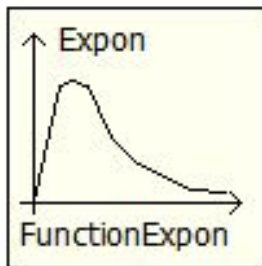
Propiedades	
Clave	Valor
Clase	FunctionDist
Nombre	FunctionDist
Etiqueta	T.Cliente
Distribucion	{30%;20%;50%}
NombreDist	TipoCliente
Semilla	3

$$\text{TipoCliente} = f(\text{RN3})$$

TipoCliente

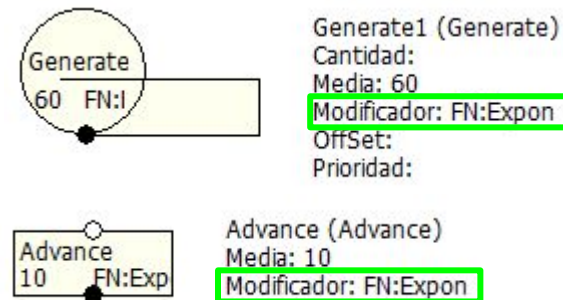


# Función Exponencial



FunctionExpon (FunctionExpon)  
NombreFunc = Expon  
Seed = 2

Se usa como **modificador** en:



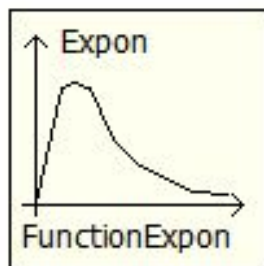
**Multiplicando la Media** por el valor de la función exponencial

# Función Exponencial



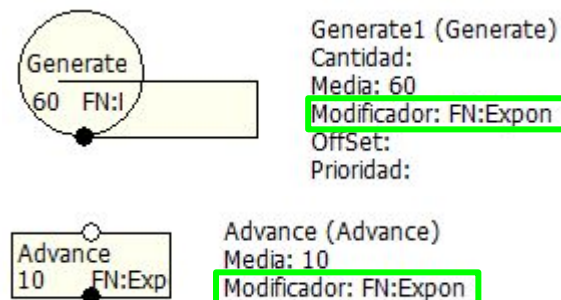
Expon      FUNCTION      RN2,C24

0,0/0.1,0.104/0.2,0.222/0.3,0.355/0.4,0.509/0.5,0.69/0.6,0.915/0.7,1.2/0.75,1.38/0.8,1.6/0.84,1.83/0.88,2.12/0.9,2.3/0.92,2.52/0.94,2.81/0.95,2.99/0.96,3.2/0.97,3.5/0.98,3.9/0.99,4.6/0.995,5.3/0.998,6.2/0.999,7.0/0.9997,8.0



FunctionExpon (FunctionExpon)  
NombreFunc = Expon  
Seed = 2

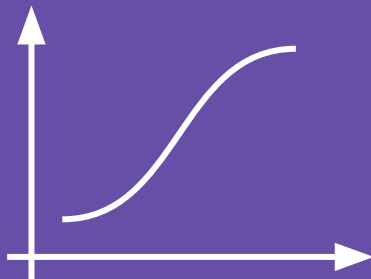
Se usa como **modificador** en:



**Multiplicando la Media** por el valor de la función exponencial

# Modelos y Simulación

Select  
Funciones



Pablo Montini  
Juan Ignacio Iturriaga  
Franco Lanzillotta