

# Tema 3. Técnicas para el diseño de casos de prueba

Auditoría, Calidad y Gestión de Sistemas  
ACG





# Análisis de Valores Límites



La técnica de valores límites es una técnica **complementaria** a las clases de equivalencia y tablas de decisión.

Busca encontrar los errores que se pueden producir en los **extremos de los datos de entrada**, ya que el sentido común indica que los puntos cercanos a los límites pueden ser más propensos a errores.

Si por ejemplo tenemos una variable de entrada que puede tomar un valor del 1 al 10, los valores límites serían el 1, el 10 y valores cercanos.



**Intervalo:** un subconjunto del espacio de los valores de entrada de un programa.

**Punto límite de un intervalo (boundary) :** aquel punto que según sea incrementado o decrementado un valor *epsilon* infinitesimal dará como resultado un valor perteneciente o no a dicho intervalo.

**Desigualdad del límite :** una expresión algebraica con  $>$ ,  $<$ ,  $\geq$ ,  $\leq$  que define parte de los puntos que pertenecen a un intervalo o dominio.

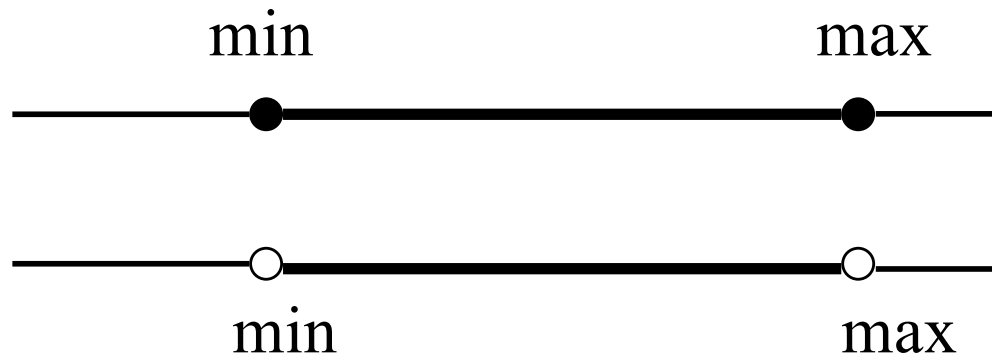
**Condiciones de igualdad :** una expresión algebraica con  $=$ ,  $\neq$



Límite cerrado = los puntos de los límites pertenecen al intervalo.

Límite abierto = los puntos de los límites no pertenecen al intervalo.

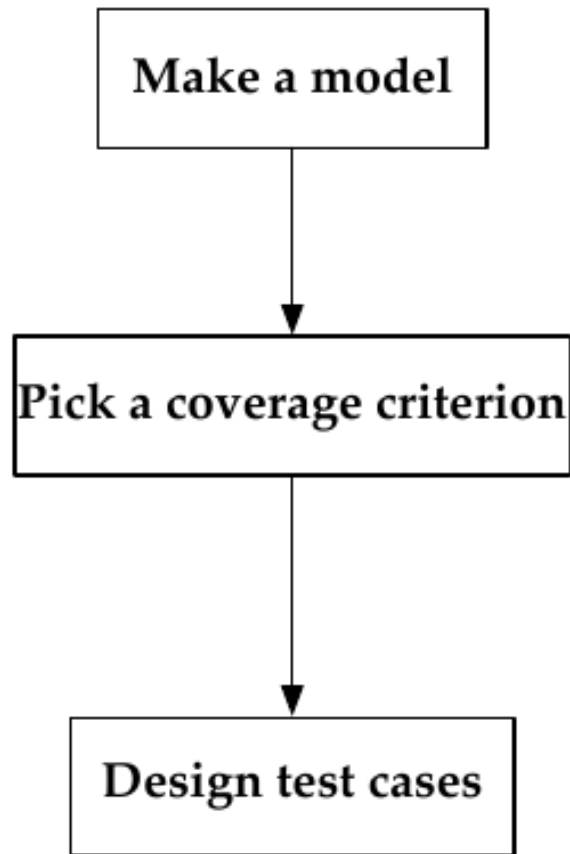
En el caso 1-dimensional



● es límite cerrado

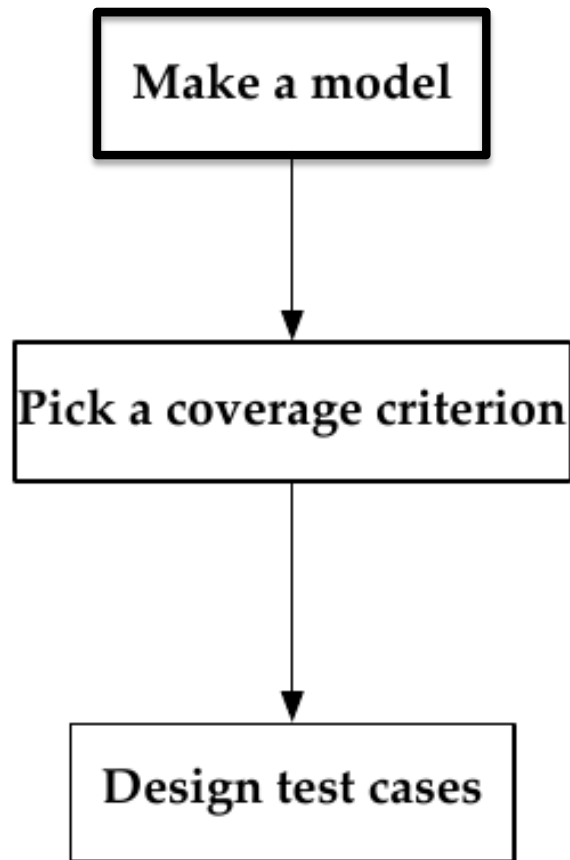
○ es límite abierto





Para analizar y  
testear todos los  
boundaries.

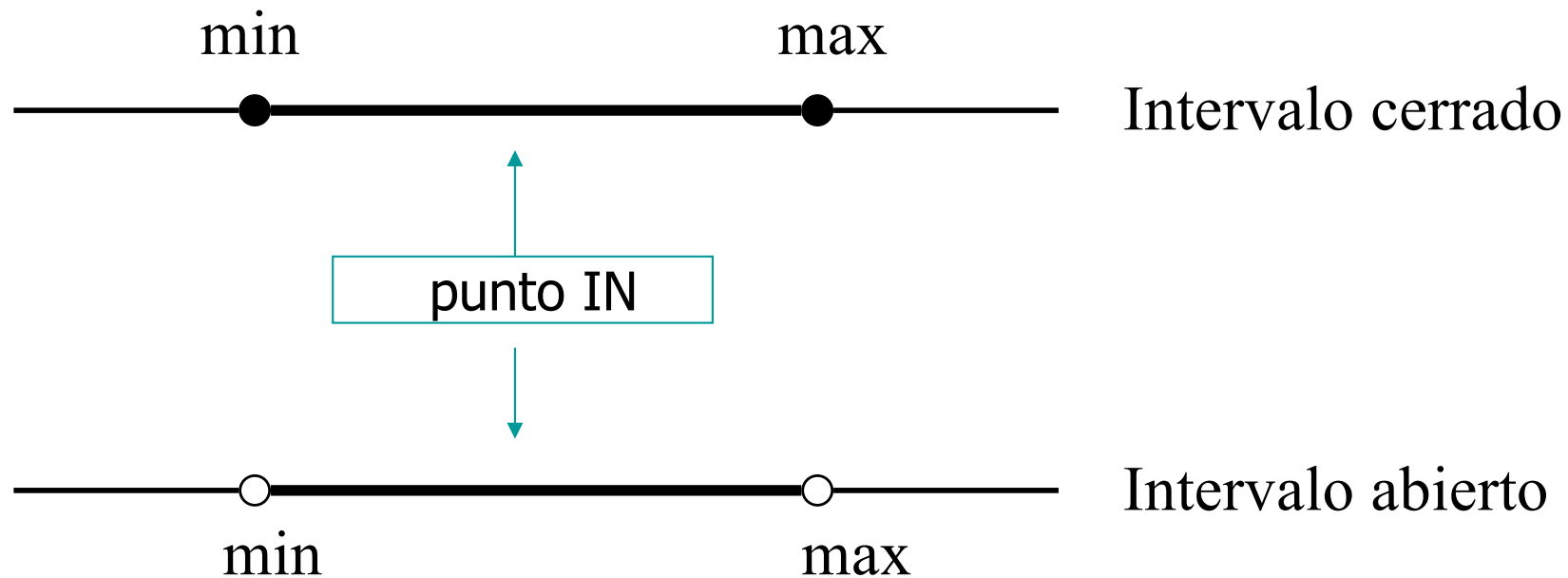




Para analizar y  
testear todos los  
boundaries.

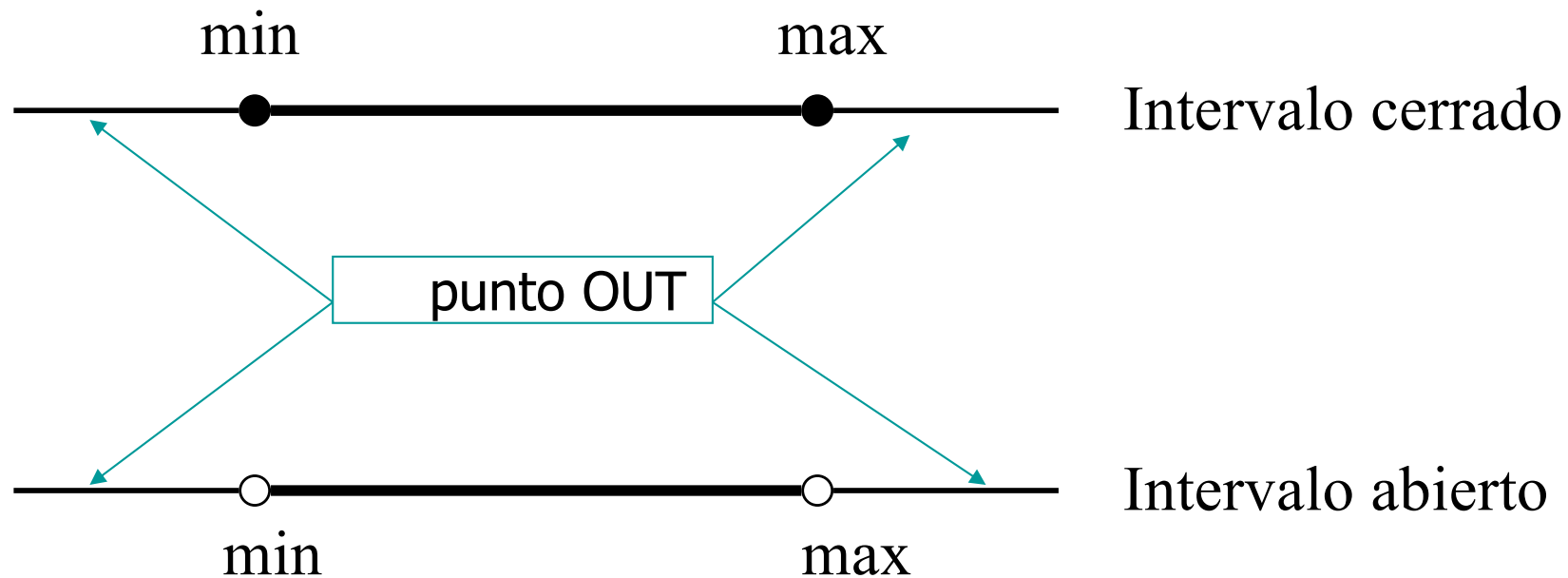


punto IN = un valor dentro del intervalo que no es un límite





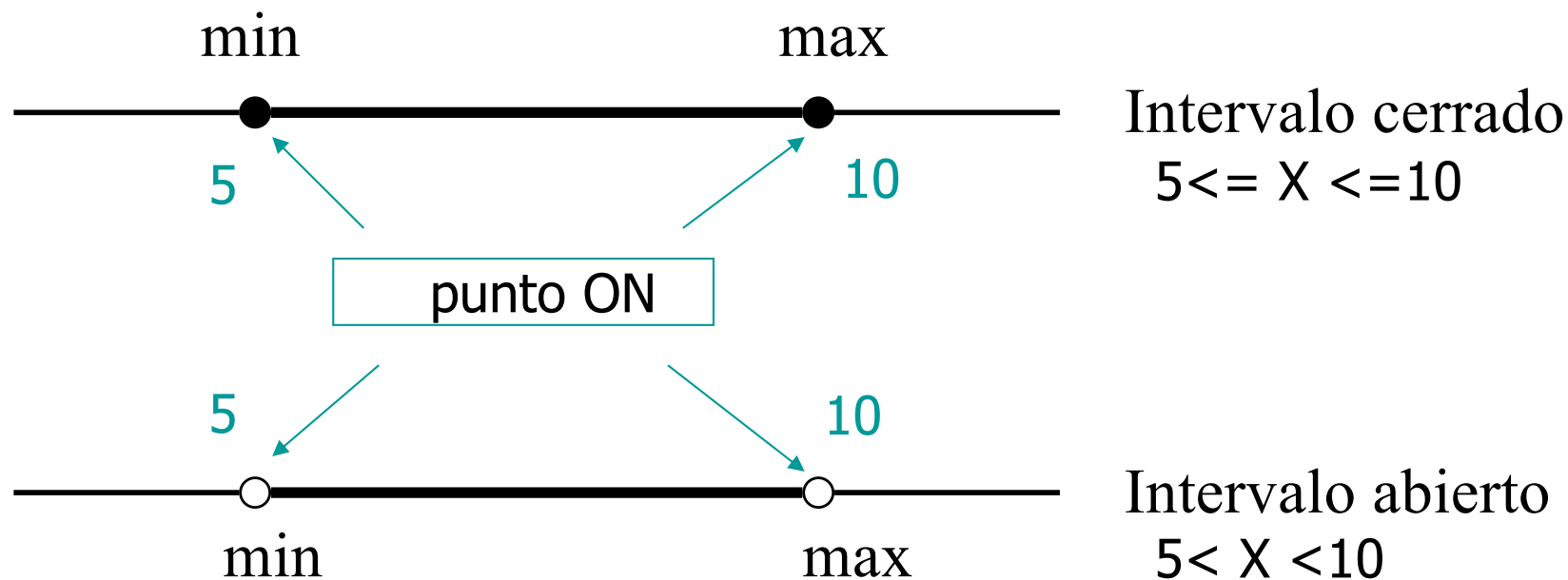
punto OUT = un valor fuera del intervalo que no es un límite



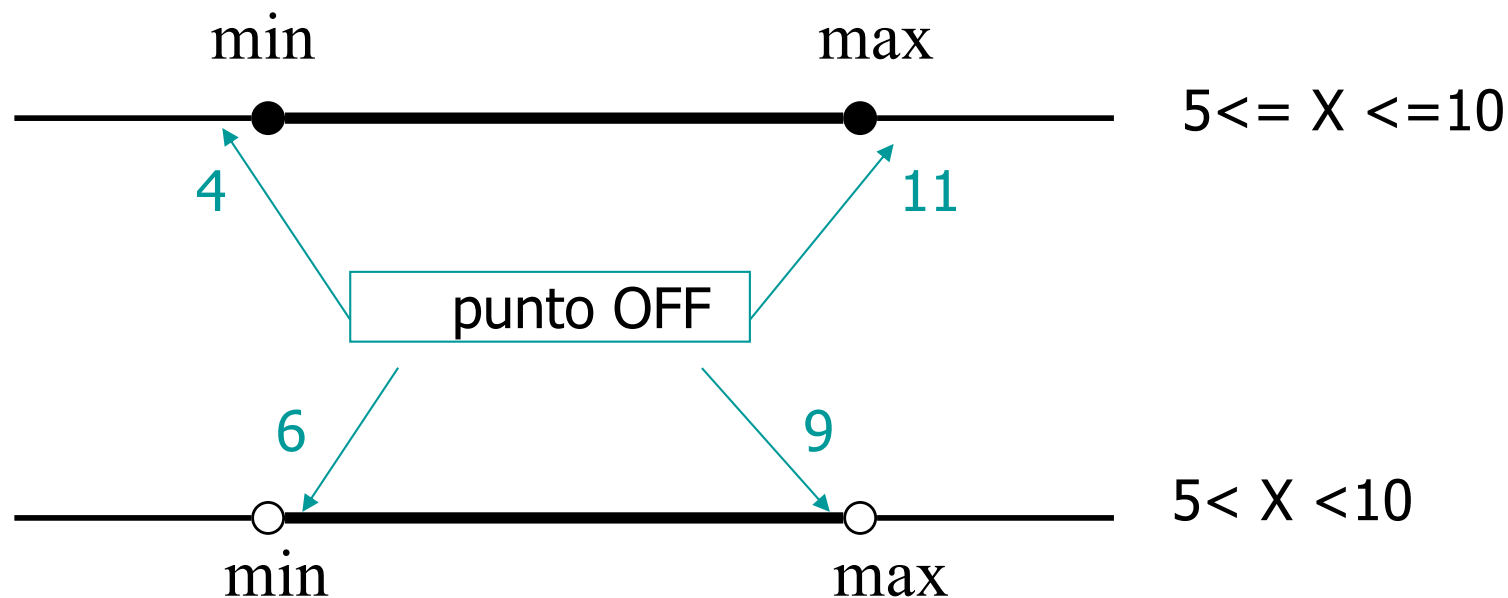
punto ON = un valor que es un punto límite de un intervalo

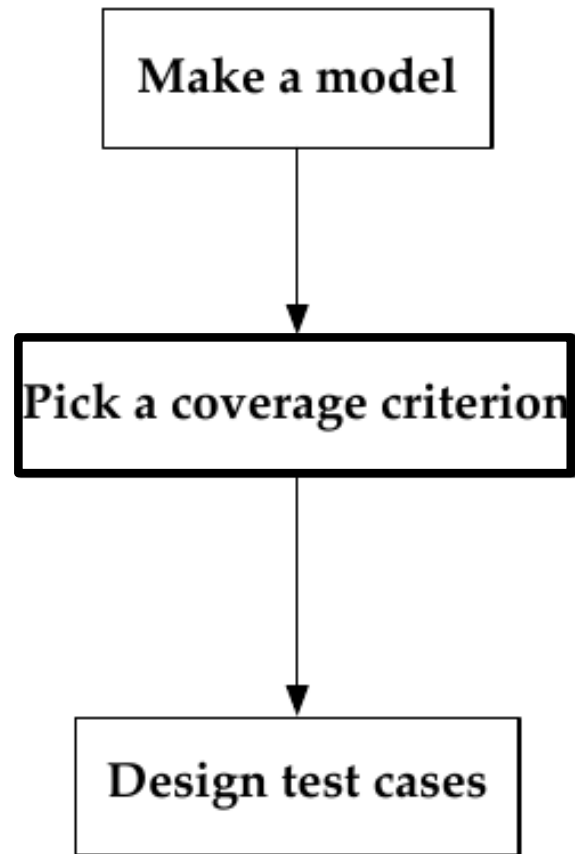
Límite cerrado (intervalo cerrado) punto ON satisface la condición del intervalo

Límite abierto (intervalo abierto) punto ON **no** satisface la condición del intervalo



punto OFF = un valor cerca del punto límite de un intervalo  
 Límite cerrado (intervalo cerrado) punto OFF **no** satisface la condición del intervalo  
 Límite abierto (intervalo abierto) punto OFF satisface la condición del intervalo





Para analizar y  
testear todos los  
boundaries.



## ESTRATEGIA:

- 1 punto ON + 1 punto OFF para cada desigualdad de límite
- Elegir los puntos tal que ON y OFF están lo más cerca posible

## Ejemplos:

- $3 \leq \text{extras} < 10$ 
  - Puntos ON: extras = 3; extras = 10
  - Puntos OFF: extras = 2; extras = 9
- $2000 < \text{precio\_base} \leq 5000$ 
  - Puntos ON??
  - Puntos OFF??



$x > 0$  (x es numero natural)

- $ON = 0$

- $OFF = 1$

$x \leq 10$  (x es numero natural)

- $ON = 10$

- $OFF = 11$

$x > 0$  (x es numero real)

- $ON = 0$

- $OFF = 0,00001$

$x \leq 10$  (x es numero real)

- $ON = 10$

- $OFF = 10,0001$



Solo hemos hablado de desigualdades

Si los límites contienen igualdades entonces habrá que elegir 2 puntos OFF:

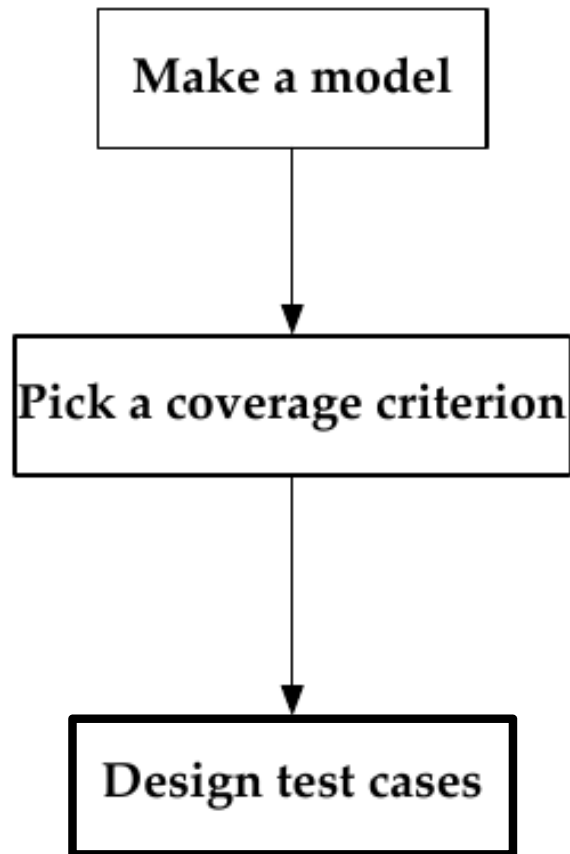
- Uno por encima del límite
- Otro por debajo

Ejemplos:

- $x=0$ , OFF (encima) = 1, OFF (debajo) = -1
- $y=1.2$ , OFF (encima) = 1.2001, OFF (debajo) = 1.1999







Para analizar y  
testear todos los  
boundaries.





Una representación  
conveniente para representar  
el conjunto de casos de test  
desarrollados con el testeo de  
dominio.

Cada columna es un caso de  
test

Cada caso de test:

- un ON o OFF para la variable  
en cuestión (diagonal)
- para las demás variables un  
IN (horizontales)

Variable/ desigualdad/tipo			Casos de testeo														
var	condición	tipo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
X1	C11	ON	■														
		OFF		■													
	C12	ON			■												
		OFF				■											
	.....	ON					■										
		OFF						■									
	C1m	ON							■								
		OFF								■							
	Típico		IN								■	■	■	■	■	■	■
	....																
	.																
Xn	Cn1	ON									■						
		OFF										■					
	.....	ON											■				
		OFF												■			
	Cnm	ON													■		
		OFF														■	
	Típico		IN	■	■	■	■	■	■	■							
	Resultado esperado																



Recordad el componente cuyos requisitos son gestionar seguro de coches según las siguientes reglas de negocio:

	Condición		Acción		
	Numero de reclamaciones	Edad	Incremento prima	Mandar aviso	Cancelar seguro
1	0	$\geq 16 \text{ y } \leq 25$	50	no	no
2	0	$> 25 \text{ y } \leq 85$	25	no	no
3	1	$\geq 16 \text{ y } \leq 25$	100	si	no
4	1	$> 25 \text{ y } \leq 85$	50	no	no
5	entre 2 y 4	$\geq 16 \text{ y } \leq 25$	400	si	no
6	entre 2 y 4	$> 25 \text{ y } \leq 85$	200	si	no
7	$\geq 5$	$\geq 16 \text{ y } \leq 85$	0	no	si



Hay 7 variantes

2 variables por variante

- ¿Cuales?

3 o 4 límites de dominio por variante

- ¿Cuáles?

Por cada variante hacemos una matriz utilizando la estrategia  
1 OFF y 1 ON

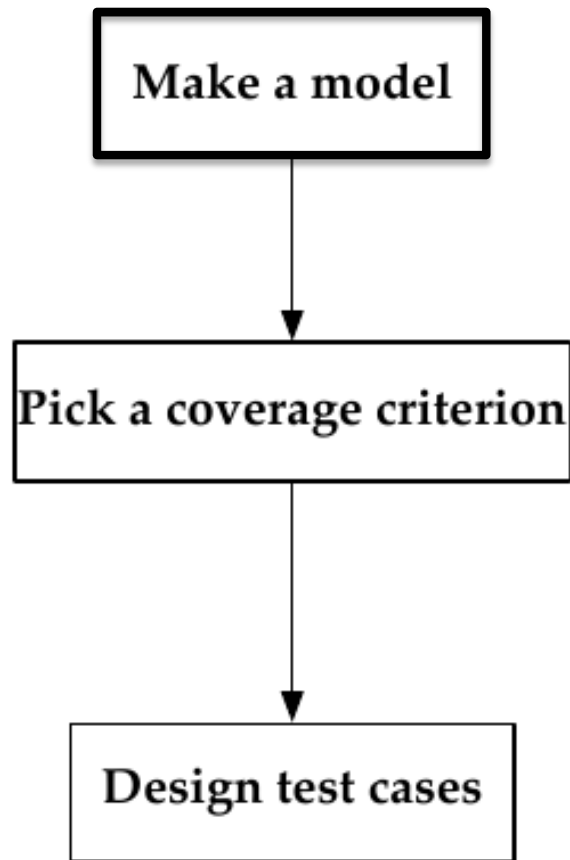


Variable/ desigualdad/tipo			Casos de testeo						
var	condición	tipo	1	2	3	4	5	6	7
nº de reclamaciones	== 0	ON	0						
		OFF		1					
		OFF			-1				
	Típico	IN				0	0	0	0
edad	≥16	ON				16			
		OFF					15		
	≤25	ON						25	
		OFF							26
	Típico	IN	20	20	20				
			Msj			Msj			
Incremento			50	100		50		50	25
Mandar aviso			no	si		no		no	no
Cancelar			no	no		no		no	no



Variable/ desigualdad/tipo			Casos de testeo							
var	condición	tipo	29	30	31	32	33	34	35	36
n° de reclamaciones	$\geq 2$	ON	2							
		OFF		1						
	$\leq 4$	ON			4					
		OFF				5				
	Típico	IN					3	3	3	3
edad	$\geq 16$	ON					16			
		OFF						15		
	$\leq 25$	ON							25	
		OFF								26
	Típico	IN	23	23	23	23				
Incremento			400	100	400	0	400	400	400	200
Mandar aviso			Sí	Sí	Sí	No	Sí	Sí	Sí	Sí
Cancelar			No	No	No	Sí	No	No	No	No





Para analizar y  
testear todos los  
boundaries.



Puntos ON y OFF para tipos de datos escalar (números, integers, reales, etc) son fáciles porque tienen orden ( $>$ ,  $<$ ,  $\leq$ ,  $\geq$ )

Pero cómo elegir puntos ON y OFF para entradas

- de tipos “nonscalar” sin orden como: strings, boolean, enumeraciones?
- de tipos complejos definidos por el usuario, en OO son objetos que son instancias de clases que representan tipos de datos complejos?





Tipos de datos “nonscalar” no tienen orden, por ejemplo:

- strings
- boolean
- enumeraciones

Los límites del dominio son cerrados y binarios:

- La variable cumple la condición o no

El Punto ON es valor que cumple la condición

El Punto OFF es valor que no cumple la condición





Algunas clases son simples y se puede tratar como tipos escalares:  
Considera la clase: Money

Imagina que tenemos que testear un componente:

- entrada un objeto Money *m*
- el componente espera que
  - $m.amount() \geq 0$
  - $m.currency() \in \{\text{EURO, USD, GBP, YEN}\}$
- Si el cliente quiere pagar en USD, GBP o YEN, tiene que pagar 10% de costes de administración
- Si el cliente paga en EURO, no hay costes de administración

```
class Money {  
    private int fAmount;  
    private String fCurrency;  
    public Money(int amount, String  
currency) {  
        fAmount= amount;  
        fCurrency= currency;  
    }  
    public int amount() {return fAmount;  
    }  
    public String currency() {return  
fCurrency;  
    }  
}
```



Variable/ desigualdad/tipo			Casos de testeo							
	condición	tipo	1	2	3	4	5	6	7	8
m.amount()	m.amount() $\geq 0$	ON	0		0					
		OFF		-1		-1				
	Típico	IN					25	25	25	25
m.currency()	m.currency() = EURO	ON					EURO			
		OFF						USD		
	Típico	IN	EURO	EURO						
	m.currency() in {USD, GBP, YEN}	ON							GBP	
		OFF								EURO
	Típico	IN			YEN	YEN				
Resultado esperado			0%	Err	10%	Err	0%	10%	10%	0%



Sin embargo hay clases que son complejas y NO se pueden tratar como escalares.

Para estas clases tenemos que utilizar la “abstracción de estados” con el objetivo de definir los límites del dominio.

Por ejemplo, imagina una clase para una *pila (stack)*. Los objetos de la clase pila pueden estar en 3 estados:

- vacío:  $(\text{stack.size()} == 0)$
- cargado:  $(\text{stack.size()} > 0 \ \&\& \ \text{stack.size()} < \text{MAXSTACK})$
- lleno:  $(\text{stack.size()} == \text{MAXSTACK})$

Utilizamos estos estados abstractos para definir los límites del dominio



Los objetos de la clase stack pueden estar en 3 estados:

- vacío: `(stack.size() == 0)`
- cargado: `(stack.size() > 0 && stack.size() < MAXSTACK)`
- lleno: `(stack.size() == MAXSTACK)`

Las condiciones de límites ahora están expresadas en estos estados  
punto ON de una condición de límite de estados = un estado que:

- Cumple con la condición
- El cambio más pequeño hace que ya no se cumpla la condición

Por ejemplo:

- Punto ON de vacío es `size()==0`
- Punto ON de cargado ???
- Punto ON de lleno ???



Los objetos de la clase stack pueden estar en 3 estados:

- vacío:  $(\text{stack.size()} == 0)$
- cargado:  $(\text{stack.size()} > 0 \ \&\& \ \text{stack.size()} < \text{MAXSTACK})$
- lleno:  $(\text{stack.size()} == \text{MAXSTACK})$

punto OFF de una condición de limite de estados = un estado que:

- No cumple con la condición
- El cambio más pequeño hace que si cumple con la condición

Por ejemplo:

- punto OFF de vacío es  $\text{size()}=1$
- punto OFF de lleno es  $\text{size()} = \text{MAXSTACK}-1$
- punto OFF de cargado es ???



Los objetos de la clase stack pueden estar en 3 estados:

- vacío:  $(\text{stack.size()} == 0)$
- cargado:  $(\text{stack.size()} > 0 \ \&\& \ \text{stack.size()} < \text{MAXSTACK})$
- lleno:  $(\text{stack.size()} == \text{MAXSTACK})$

punto IN de una condición de limite de estados = un estado que:

- Cumple con la condición
- Que no es un punto ON ni OFF (si es posible)

Por ejemplo:

- punto IN de vacío es  $\text{size()}=0$  (tambien punto ON)
- punto IN de lleno es  $\text{size()} = \text{MAXSTACK}$  (tambien punto ON)
- punto IN de cargado es ???







# Modelo de máquinas de estado



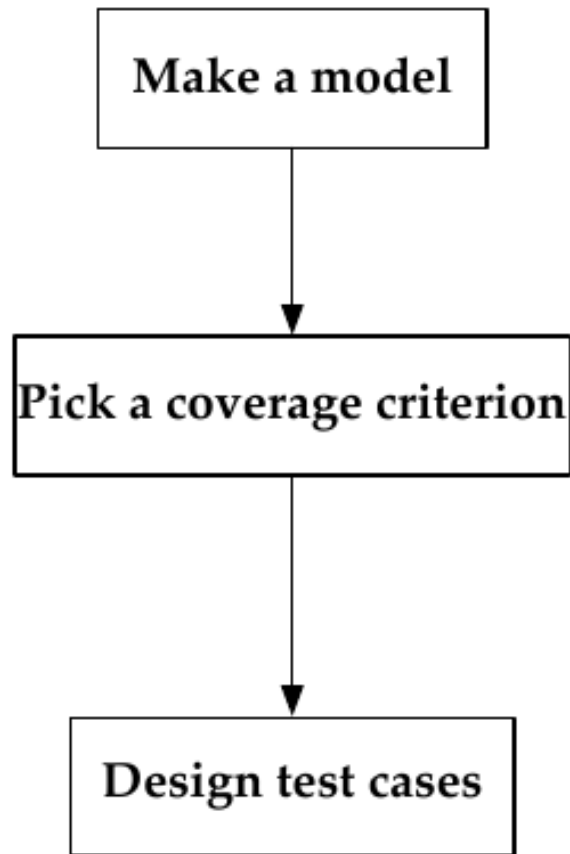
Otra técnica de diseño de casos de prueba basada en modelos.

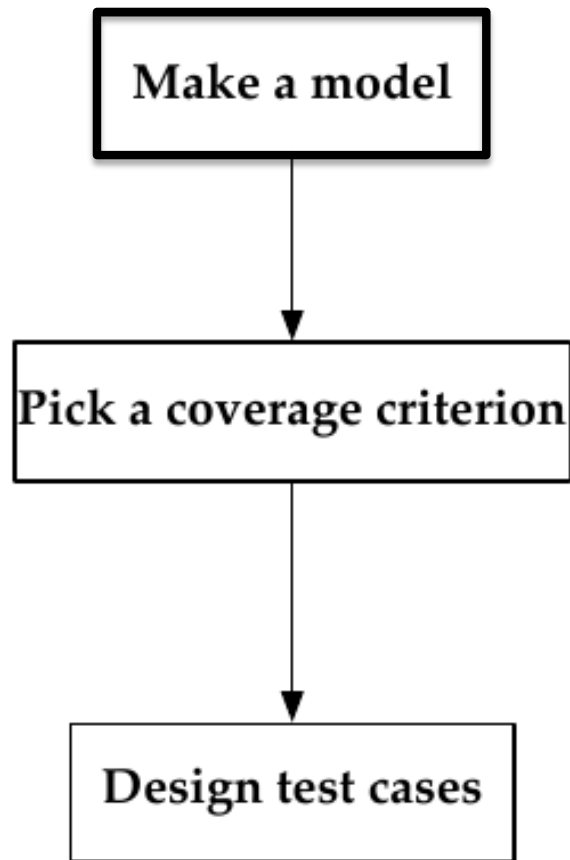
La idea es modelar el comportamiento del sistema como una máquina de estados.

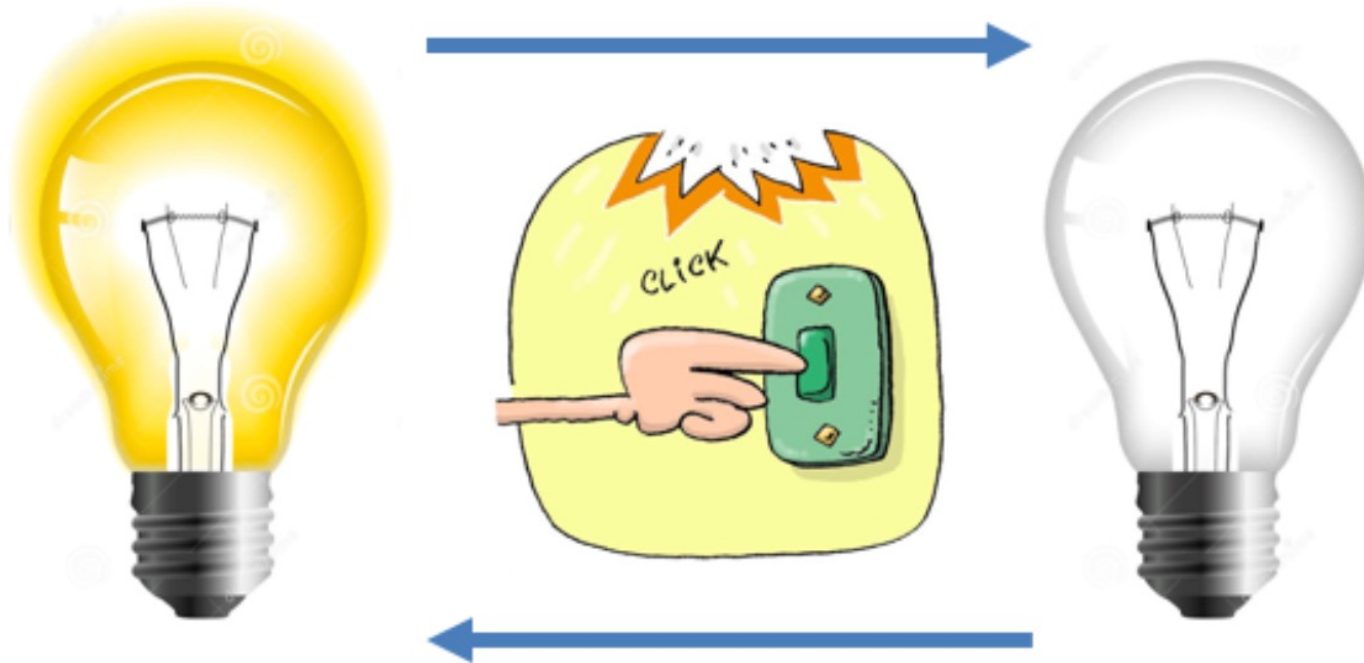
Los casos de prueba diseñados con esta técnica tratarán de cubrir los diferentes estados y transiciones especificados en la máquina de estados que representa el sistema.

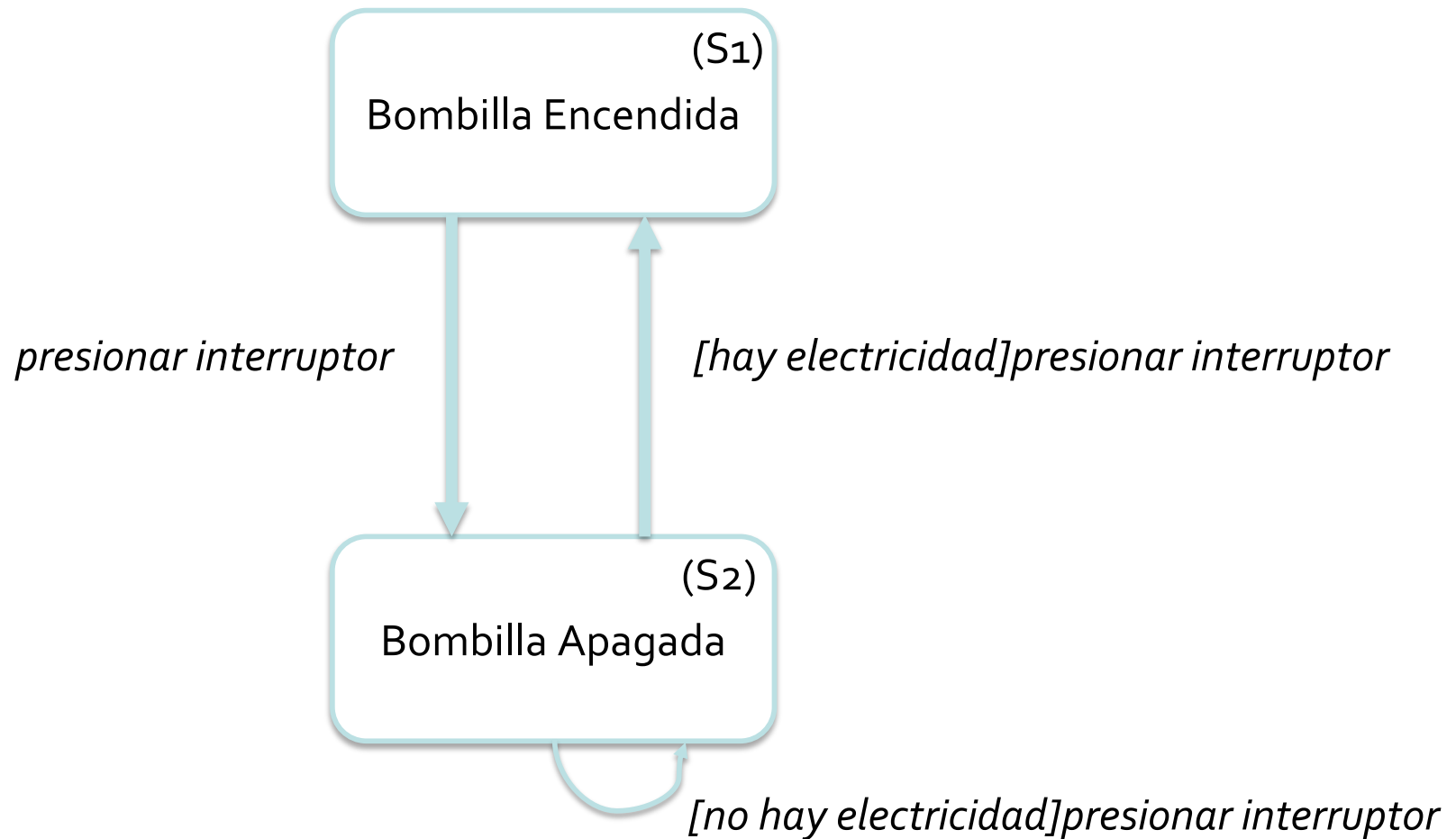










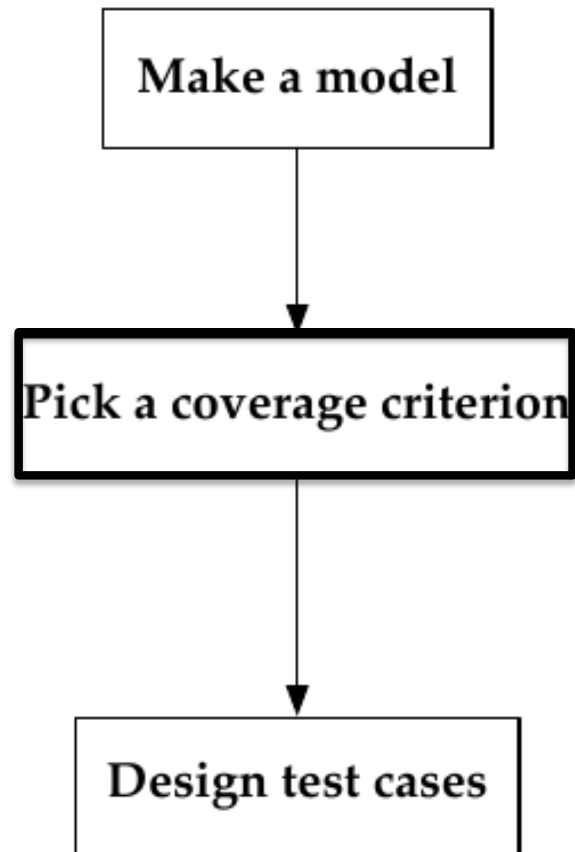


	1	2	3
Estado inicial	S1 (encendido)	S2 (apagada)	S2 (apagada)
Entrada	Presionar interruptor	Presionar interruptor	Presionar interruptor
Condición	[Hay electricidad]	[Hay electricidad]	[No hay electricidad]
Estado final	S2 (apagada)	S1 (encendido)	S2 (apagada)



- Caso de prueba 1: Lámpara apagada, sin electricidad, presiono el interruptor.
- Caso de prueba 2: Lámpara apagada, con electricidad, presiono el interruptor.
- Caso de prueba 3: Lámpara encendida, presiono el interruptor.







Algunos criterios de cobertura son:

- **Cobertura de estados:** Este criterio se satisface cuando los casos de prueba recorren todos los estados.
- **Cobertura de transiciones:** En este caso, se satisface el criterio cuando se recorren todas las transiciones.
- **Cobertura de pares de transiciones:** Para cada estado se cubren las combinaciones de transiciones de entrada y salida.



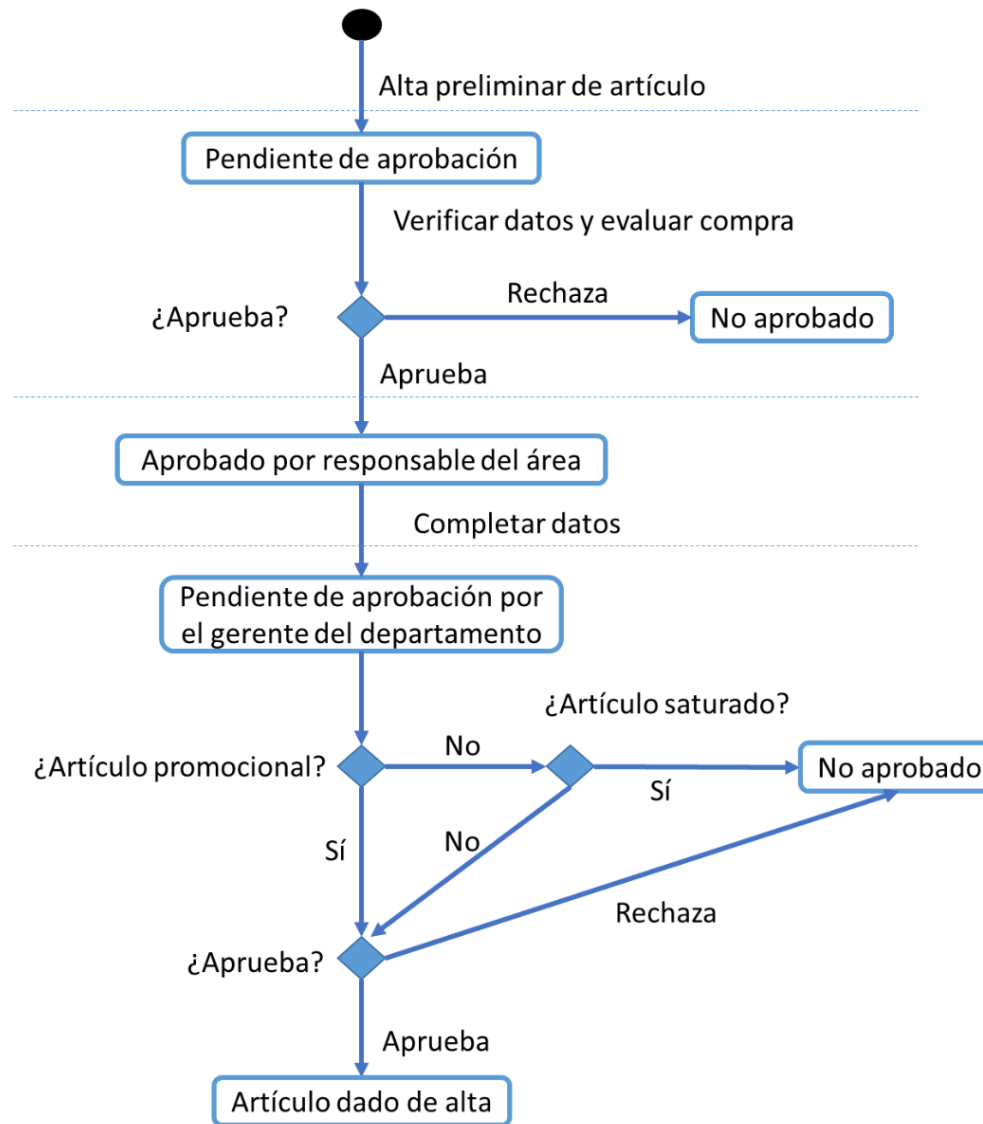


Cobertura de estados: 2

Cobertura de transiciones: 1, 2 y 3.

Cobertura pares de transiciones: 1, 2 y 3





- Caso 1: Alta preliminar de artículo, verificar datos y rechazar compra.
- Caso 2: Alta preliminar de artículo, verificar datos, aprobar compra por responsable, completar los datos por *trade marketing*, el gerente de departamento de compras aprueba la compra para un artículo promocional.
- Caso 3: Alta preliminar de artículo, verificar datos, aprobar compra por responsable completar los datos por *trade marketing*, el gerente de departamento de compras aprueba la compra para un artículo no promocional y no saturado.
- Caso 4: Alta preliminar de artículo, verificar datos, aprobar compra por responsable completar los datos por *trade marketing*, el gerente de departamento de compras no aprueba la compra por ser un artículo no promocional y saturado.
- Caso 5: Alta preliminar de artículo, verificar datos, aprobar compra por responsable, completar los datos por *trade marketing*, el gerente de departamento de compras no aprueba la compra para un artículo promocional.
- Caso 6: Alta preliminar de artículo, verificar datos, aprobar compra por responsable, completar los datos por *trade marketing*, el gerente de departamento de compras aprueba la compra para un artículo no promocional y no saturado.



Cobertura de estados: 1, 4, 5

Cobertura de transiciones: 1, 2, 3 y 5.

Cobertura pares de transiciones: 1, 2, 3, 4 y 5.

