



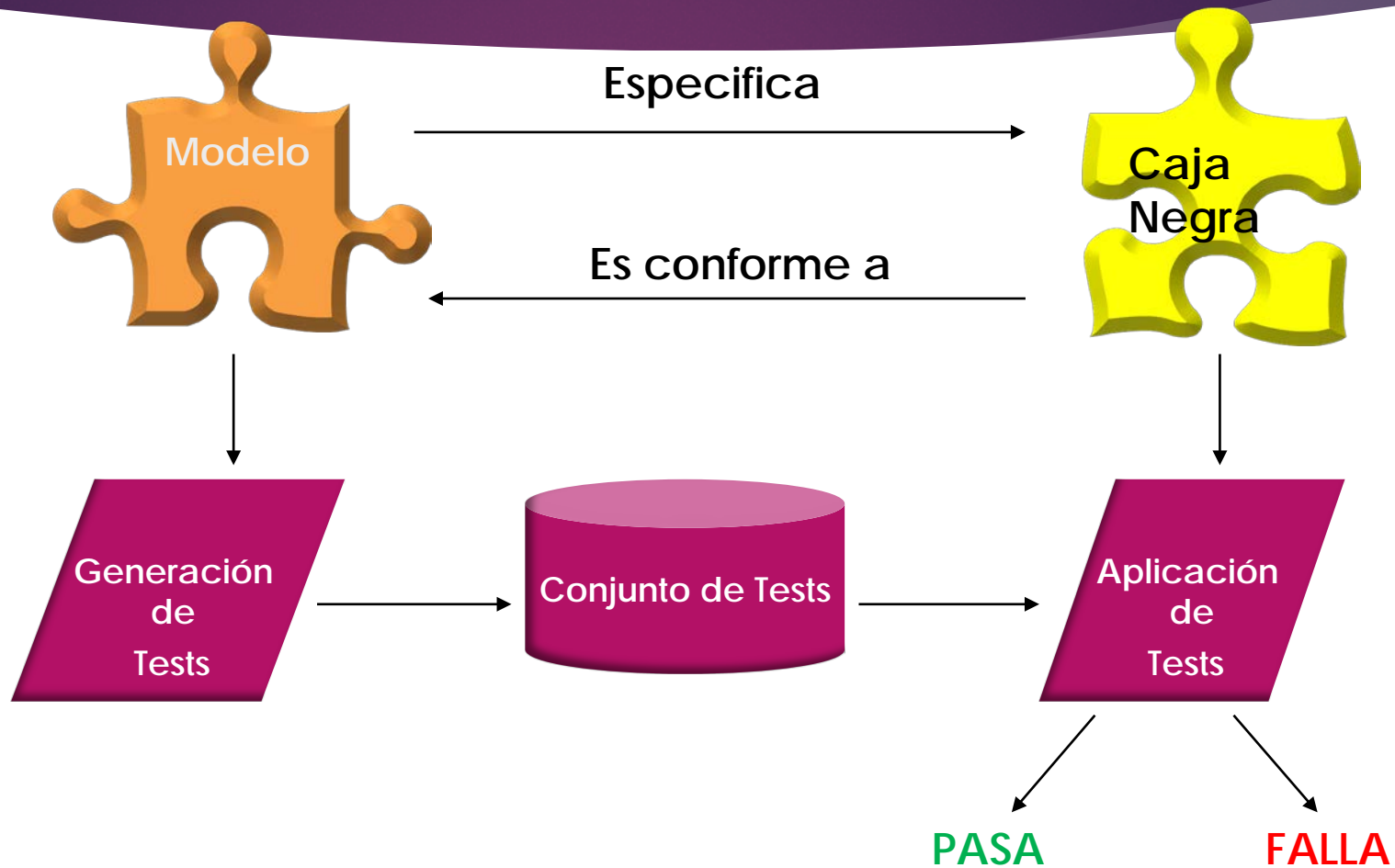
# Design and Testing of Reliable Systems

Manuel Núñez

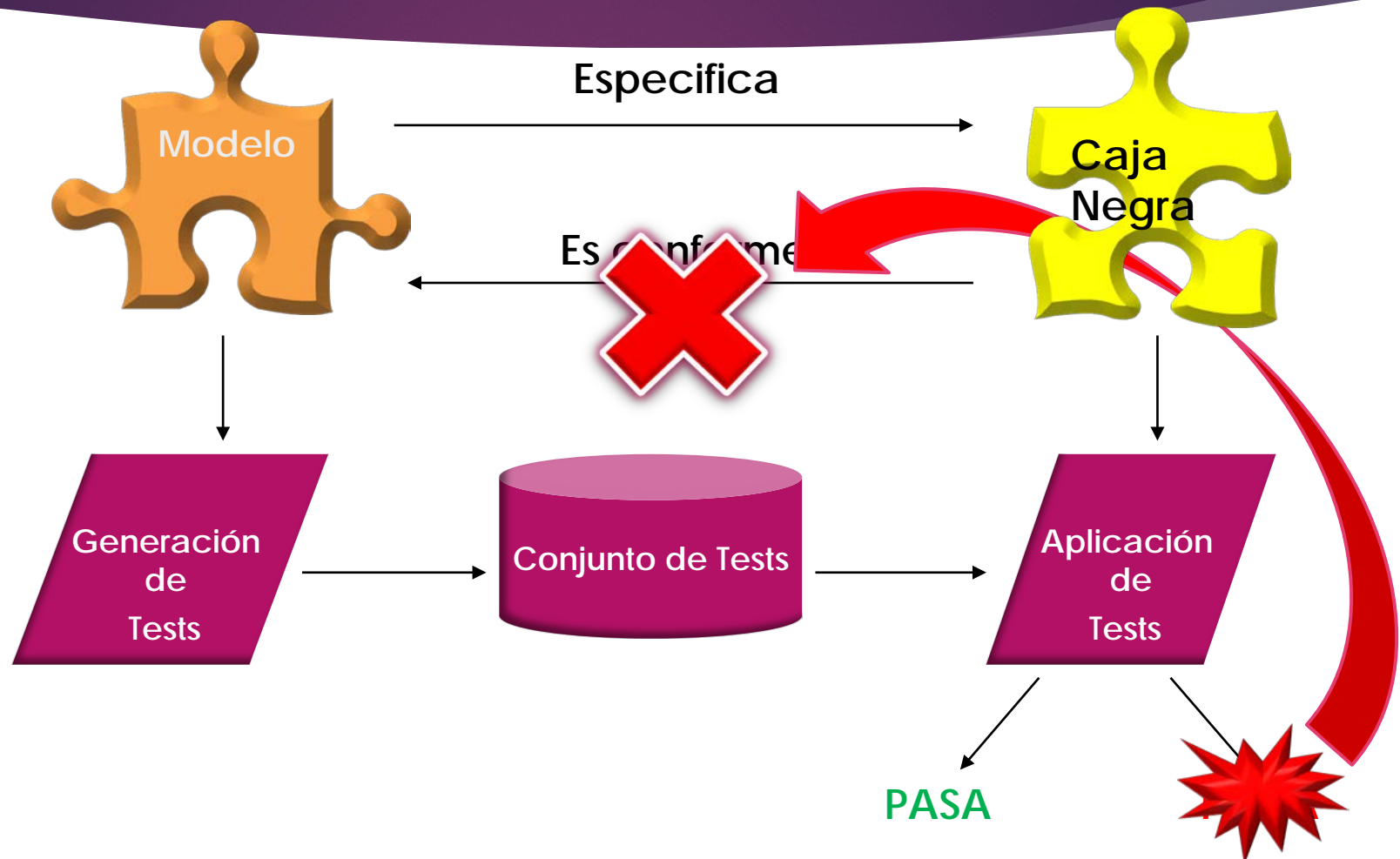
# Principales líneas de trabajo

- ▶ Conformance Testing (línea principal).
- ▶ Metamorphic testing (aplicado a cloud).
- ▶ Modelado de contratos.

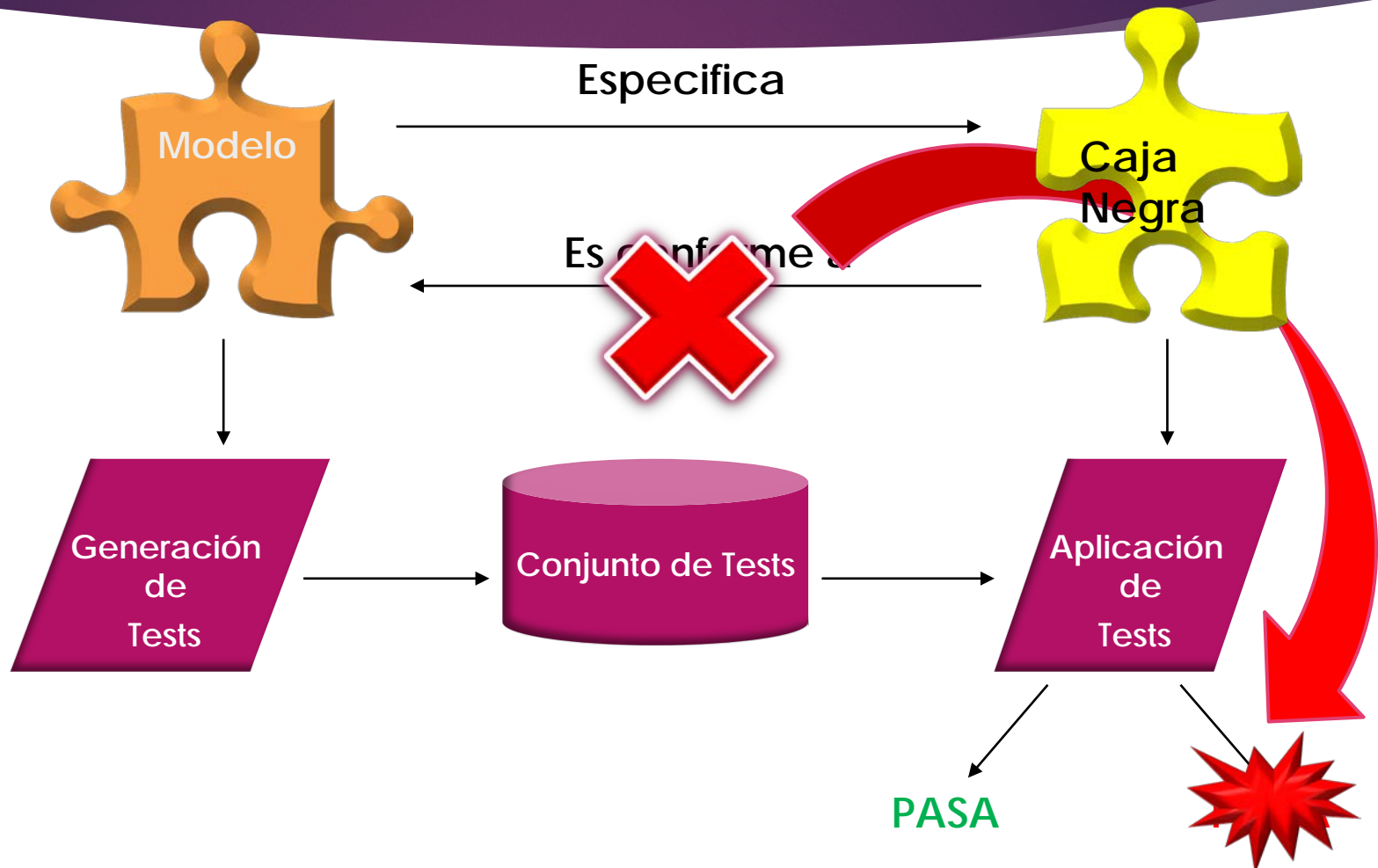
# Conformance Testing (in a nutshell)



# Conformance Testing: Corrección



# Conformance Testing: Completitud



# Conformance Testing (in two nutshells)

En mi grupo hemos desarrollado marcos (formales) de testing, junto con herramientas para ponerlos en práctica, para:

¿Qué hace mi caja negra?

**Acciones que hace el sistema.**

¿Cómo hace mi caja negra las cosas?

**Tiempo/probabilidades/recursos utilizados para hacer acciones.**

¿Puedo interaccionar con mi caja negra?

**Testing activo vs. pasivo (monitoring).**

¿Cómo está distribuido mi sistema?

**Testing distribuido, comunicaciones síncronas/asíncronas.**

¿Cómo genero *buenos* (no exhaustivos) conjuntos de tests?

**Algoritmos genéticos, uso de Information Theory.**

# Metamorphic Testing

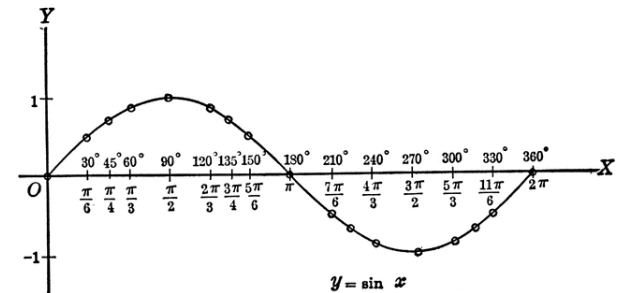
Habitualmente no tenemos un oráculo/especificación pero aún así podemos intentar testear, de forma sistemática y (semi-)formal.

¿Qué es metamorphic testing?

Tenemos que testear un sistema que compute  $\sin(x)$ .

¿Tests?  $\sin(0)$ ,  $\sin(30)$ ,  $\sin(90)$ ,  $\sin(180)$ ,  $\sin(45)$

¿Oráculo?



Ya empiezan los problemas (igualdad entre reales). ¿ $\sin(61)$ ?

Incluso aunque no sepamos el valor de  $\sin(x)$ , sabemos que es igual a  $\sin(x+360)$ ,  $\sin(180-x)$ ,  $-\sin(x+180)$ .

También sabemos  $\sin(x)^2 + \cos(x)^2 = 1$  o que  $\sin(-x) = -\sin(x)$ ....



# Metamorphic Testing

Se ha aplicado Metamorphic Testing a campos muy heterogéneos:

Simulation and modeling

Numerical analysis

Web services

Compilers

Decision support

Computer graphics

Embedded systems

Machine learning

Components

RESTful Web APIs

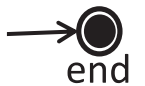
Bioinformatics



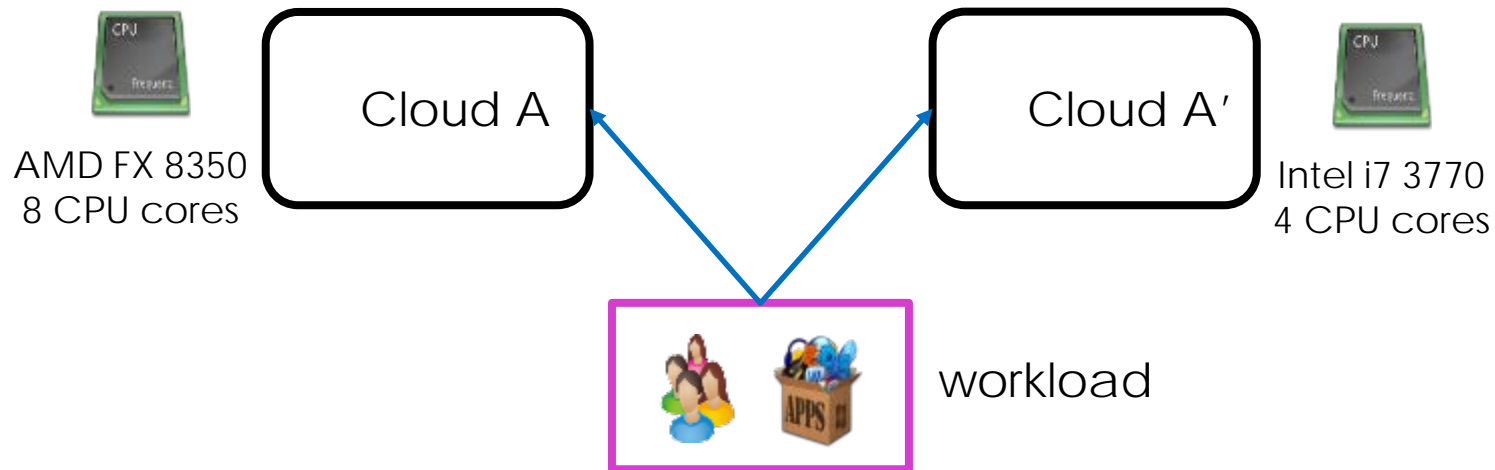
# MT (in a nutshell)



yes]



# MT en DTRS: Cloud computing



MR: Un Sistema más lento necesitará más tiempo para ejecutar la misma carga de trabajo.

Cambiamos todas las CPUs del cloud A por otras más lentas.

Si A' is más lento que A  $\rightarrow$  tiempo de ejecución A < tiempo de A'

# MT en DTRS: Cloud computing

Se ha combinado MT (para generar *follow up tests*) con algoritmos evolutivos para evolucionar clouds y optimizar configuraciones.

Trabajo actual/futuro:

- Combinar MRs para mejorar la efectividad del testing.
- Identificar (semi-automáticamente) buenas MRs.
- Optimizar sistemas cloud: aspectos no funcionales (performance, energía, coste).

# Análisis formal de contratos

## Extracto de las regulaciones en el aeropuerto de Barajas (lenguaje natural)

1. The passenger is permitted to use the check-in desk within two hours before the plane takes off ( $t_0$ ).
2. At the check-in desk, the passenger is obliged to present her boarding pass within 5 minutes.
3. After presenting the boarding pass, the passenger must show her passport, she has 5 minutes for this purpose.
4. Henceforth, the passenger is (i) prohibited from carrying liquids in her hand-luggage until boarding; and (ii) prohibited from carrying weapons during the whole trip until the plane lands. If she has liquids in her hand-luggage, she is obliged to dispose of them within 10 minutes.
5. After presenting her passport, the passenger is permitted to board within 90 minutes and to present the hand-luggage to the staff within 10 minutes. Therefore, the airline company is obliged to allow the passenger to board within 90 minutes. If the passenger is stopped from carrying luggage, the airline company is obliged to put the passenger's hand luggage in the hold within 20 minutes.

# Análisis formal de contratos: DTRS

Extracto de las regulaciones en el aeropuerto de Barajas  
(lo mismo pero con fórmulas)

$$\begin{aligned}\varphi_0 &::= \mathcal{P}_p(\text{checkin})[t_0 - 120] \\ \varphi_1 &::= \mathcal{O}_p(\text{PBP})[5] \\ \varphi_2 &::= \mathcal{O}_p(\text{ShP})[5] \\ \varphi_3 &::= (\mathcal{F}([\text{weapon}, p] \mathcal{U} [\text{landing}, c])) \wedge \\ &\quad ((\mathcal{F}([lq, p] \mathcal{U} [\text{boarding}, p])) \blacktriangleright \mathcal{O}_p(\text{dlq})[10]) \\ \varphi_4 &::= (\mathcal{P}_p(\text{brd})[90]; \mathcal{P}_p(\text{h1})[10]) \blacktriangleright \\ &\quad (\mathcal{O}_c(\text{brd})[90]; \mathcal{O}_c(\text{h1h1d})[20]) \\ PBS &::= \varphi_0; \varphi_1; \varphi_2; (\varphi_3 \wedge \varphi_4)\end{aligned}$$

# Análisis formal de contratos: DTRS

- Basado en lógica deóntica: permisos, obligaciones y prohibiciones.
- Recuperación tras fallos.
- Restricciones temporales.
- Condicionales.
- Recursión.

$$\begin{aligned} \varphi \quad ::= & \top \mid \perp \mid \mathcal{P}_k(a)[d] \mid \mathcal{O}_k(a)[d] \mid \mathcal{F}_k(a)[d] \\ & \mid \text{wait}(d) \mid \text{cond}_k(a)[d](\varphi_1, \varphi_2) \mid \varphi_1; \varphi_2 \\ & \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \blacktriangleright \varphi_2 \mid \text{rec } x.\varphi \mid x \end{aligned}$$

# Análisis formal de contratos: DTRS

- Semántica operacional de contratos.
- Composición de sistemas.
- Definición formal de incumplimiento de contratos.
- Equivalencia de contratos basado en relaciones de simulación.





Fin

# Gracias por vuestra atención

¿Preguntas?

?Comentarios?