



Design and Testing of Reliable Systems

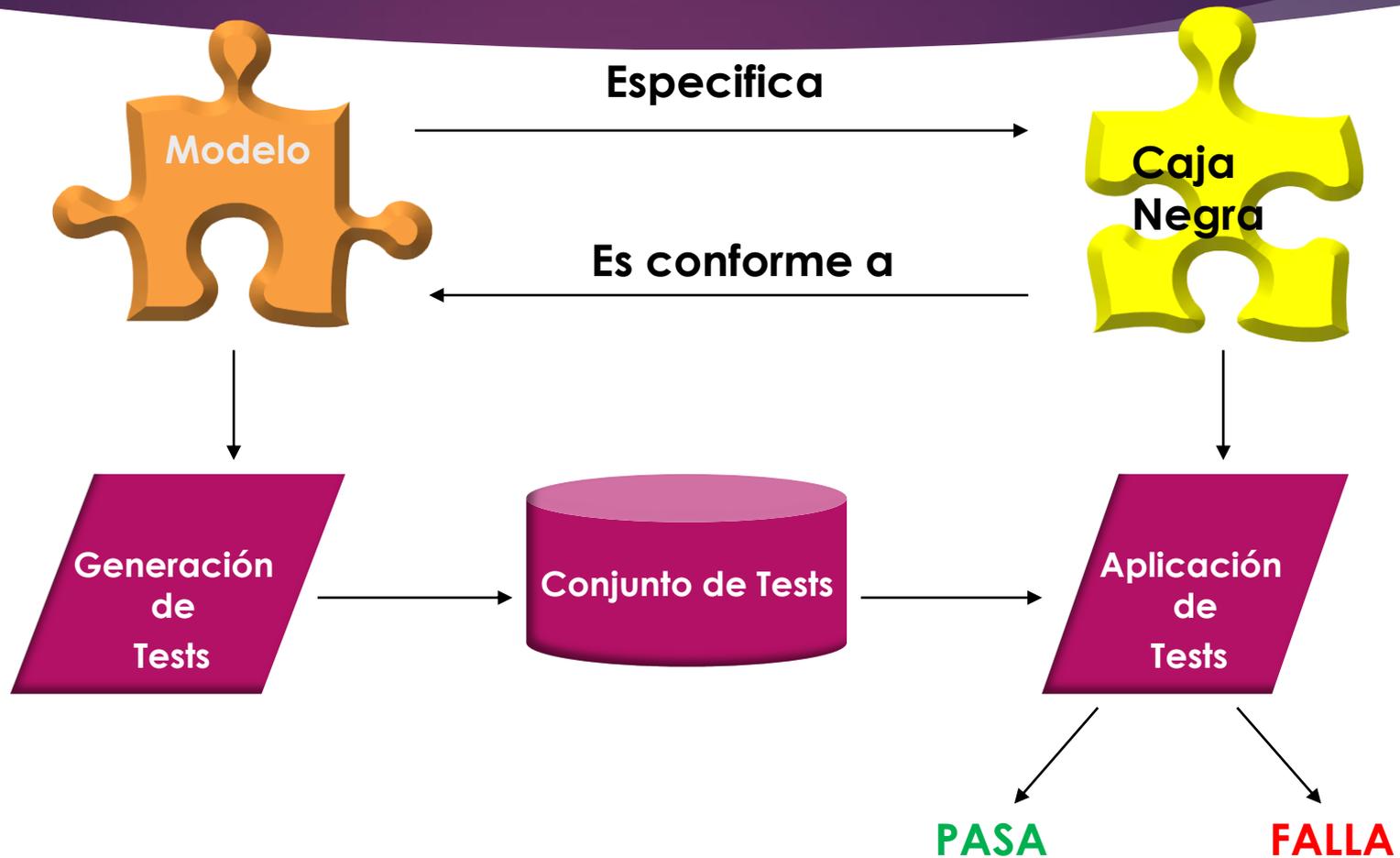
**Universidad Complutense
de Madrid**

Manuel Núñez

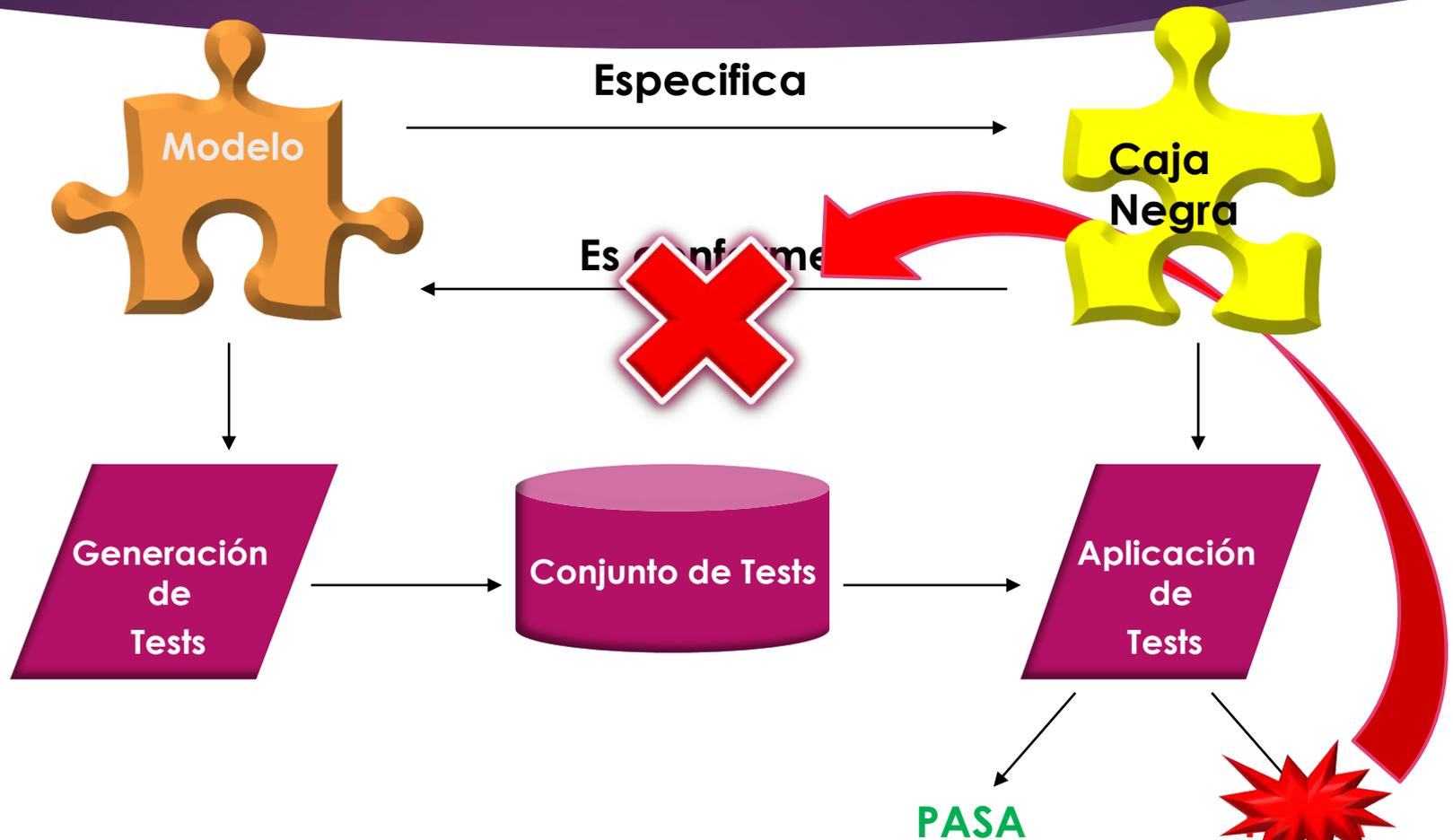
Principales líneas de trabajo (relacionadas con SEBASNet)

- ▶ Conformance Testing (línea principal).
- ▶ **Aproximaciones heurísticas en testing.**
- ▶ Metamorphic testing (aplicado a cloud).

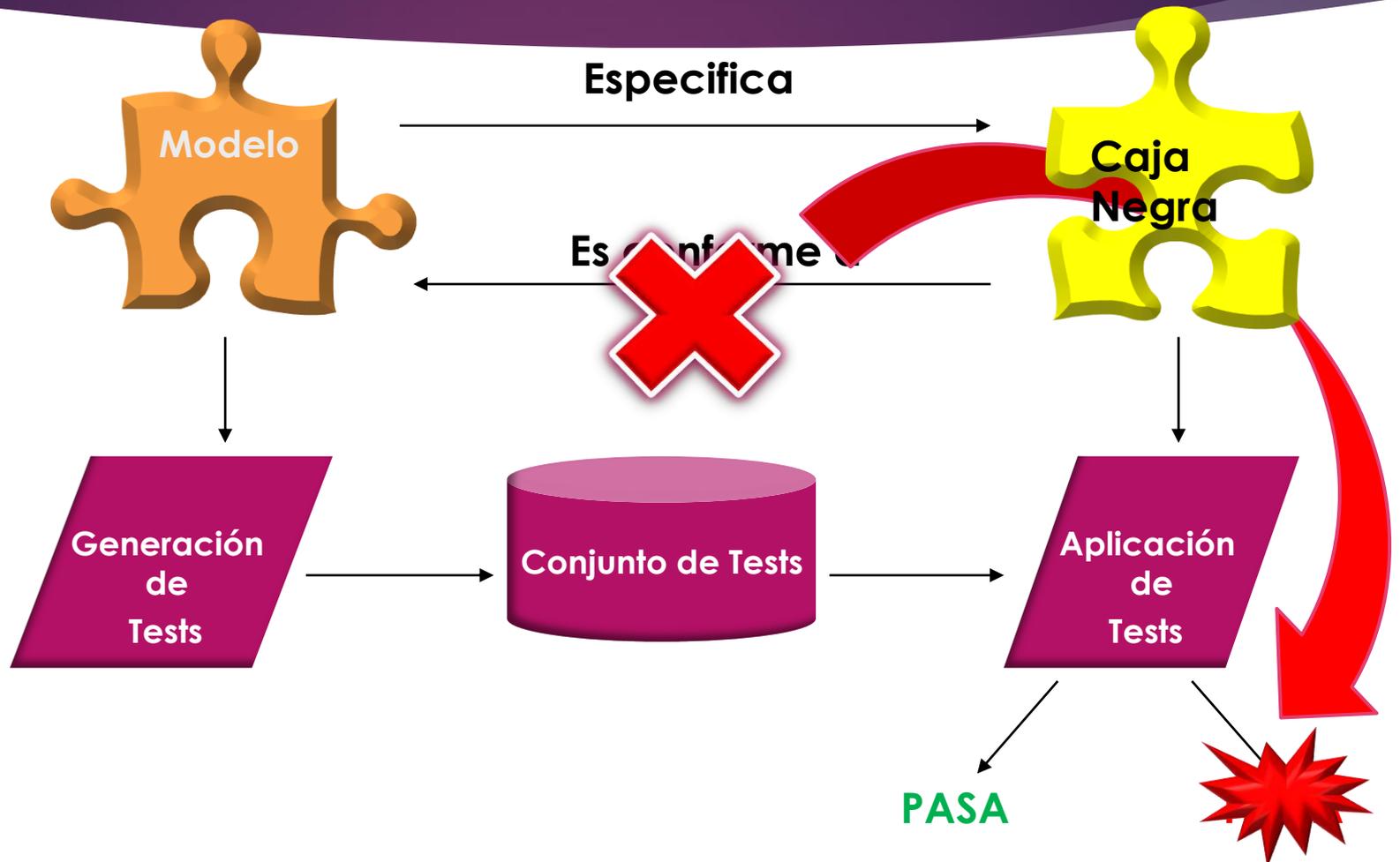
Conformance Testing (in a nutshell)



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*.

Seleccionando *buenos* tests

A veces hay más tests de lo que nuestro presupuesto o tiempo permite ejecutar. Por eso hay que elegir un *buen* subconjunto de tests.

Usualmente nos apoyamos en *mutation testing*: elegir los tests que más mutantes matan usando un número *pequeño* de inputs.

Seleccionando *buenos tests*

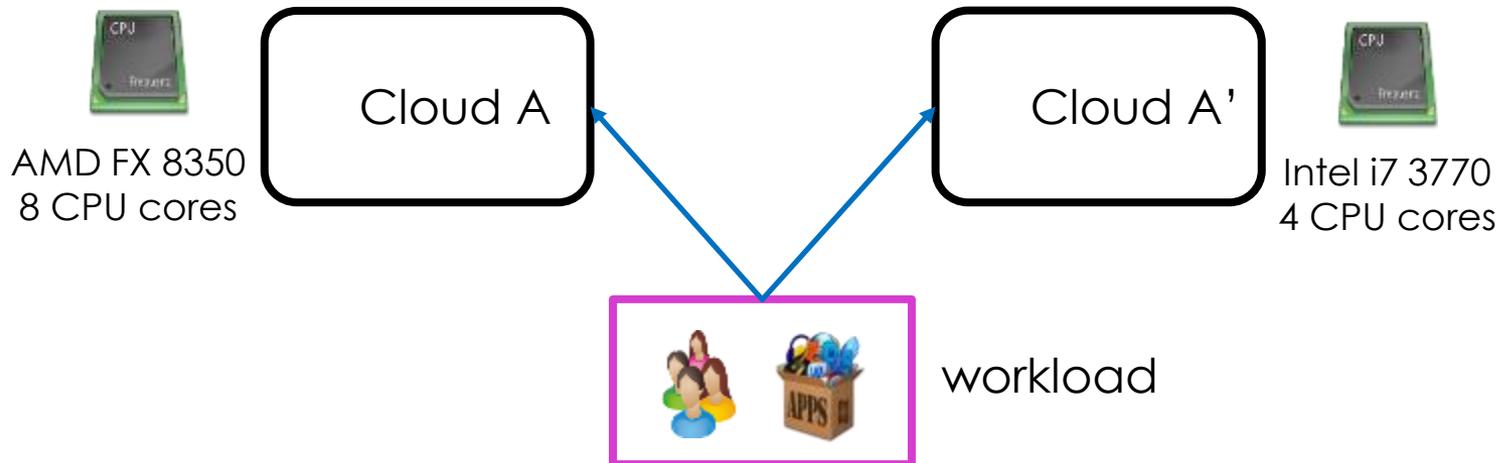
Usamos diferentes heurísticas para escoger dichos tests:

- ▶ Algoritmos genéticos y de programación genética.
- ▶ Algoritmos de enjambre.

Usamos *Teoría de la Información* para escoger tests con unas ciertas características y analizar sistemas antes de testing.

- ▶ *Mutual Information* para buscar tests que revelen casos de *Failed Error Propagation*.
- ▶ *Squeeziness* para evaluar predisposición a *Failed Error Propagation*.

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).

Fin

Gracias por vuestra atención

¿Preguntas?

¿Comentarios?