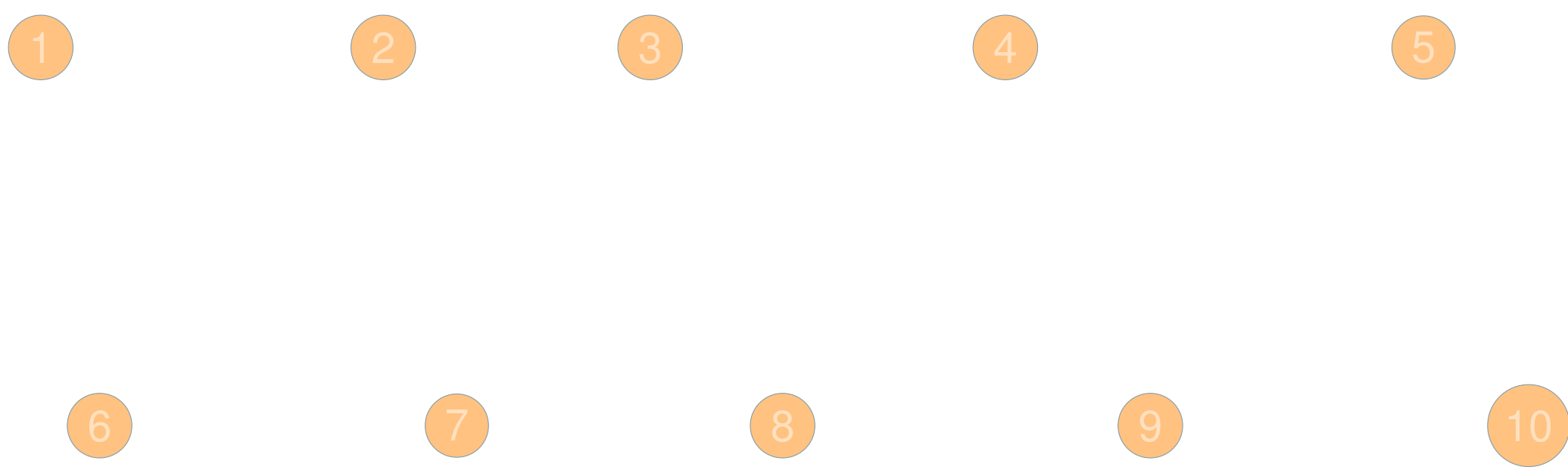


Miembros

Proyectos en ejecución



Responsable

Inmaculada Medina Buló<sup>8</sup>

Investigadores

Antonia Estero Botaro<sup>9</sup>  
Antonio García Domínguez<sup>4</sup>  
Francisco Palomo Lozano<sup>5</sup>  
Guadalupe Ortiz Bellot<sup>10</sup>  
José Antonio Jiménez Millán<sup>2</sup>  
Juan Boubeta Puig<sup>1</sup>  
Juan José Domínguez Jiménez<sup>3</sup>  
Lorena Gutiérrez Madroñal<sup>7</sup>  
M<sup>a</sup> del Carmen de Castro Cabrera<sup>6</sup>

Propios

TIN2011-27242 «Extensión de una metodología dirigida por modelos para SOA 2.0: prueba y adaptación de servicios». *MICINN*.  
PR2011-004 «Verificación, validación y adaptación en arquitecturas orientadas a servicios aplicando una metodología dirigida por modelos». *UCA*.

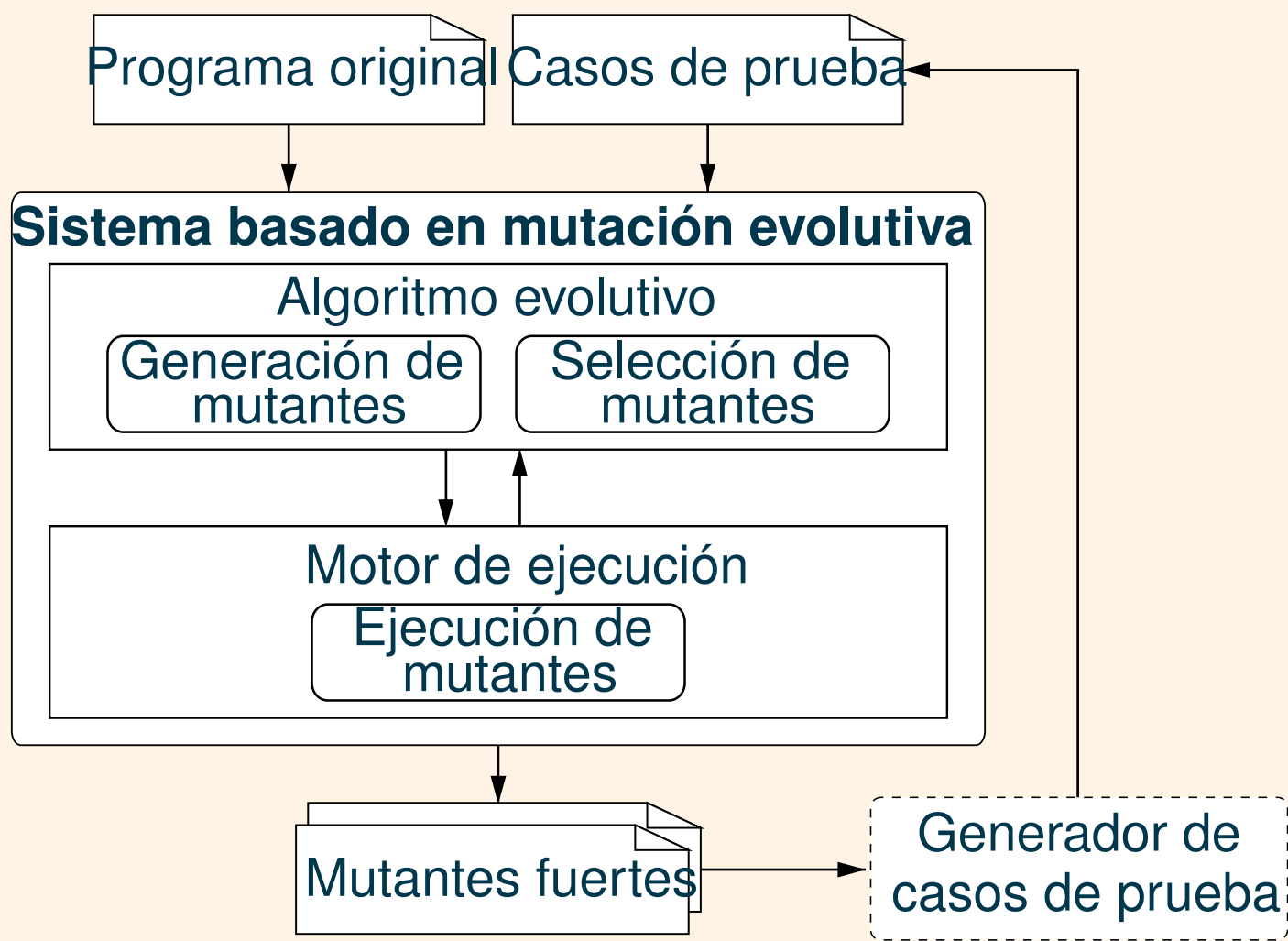
En colaboración

TIN2008-02985 «Desarrollo dirigido por modelos de procesos de negocio en factorías software: aplicaciones a la Web 2.0 y arquitecturas multicapa en J2EE». *MICINN*.

Líneas de investigación

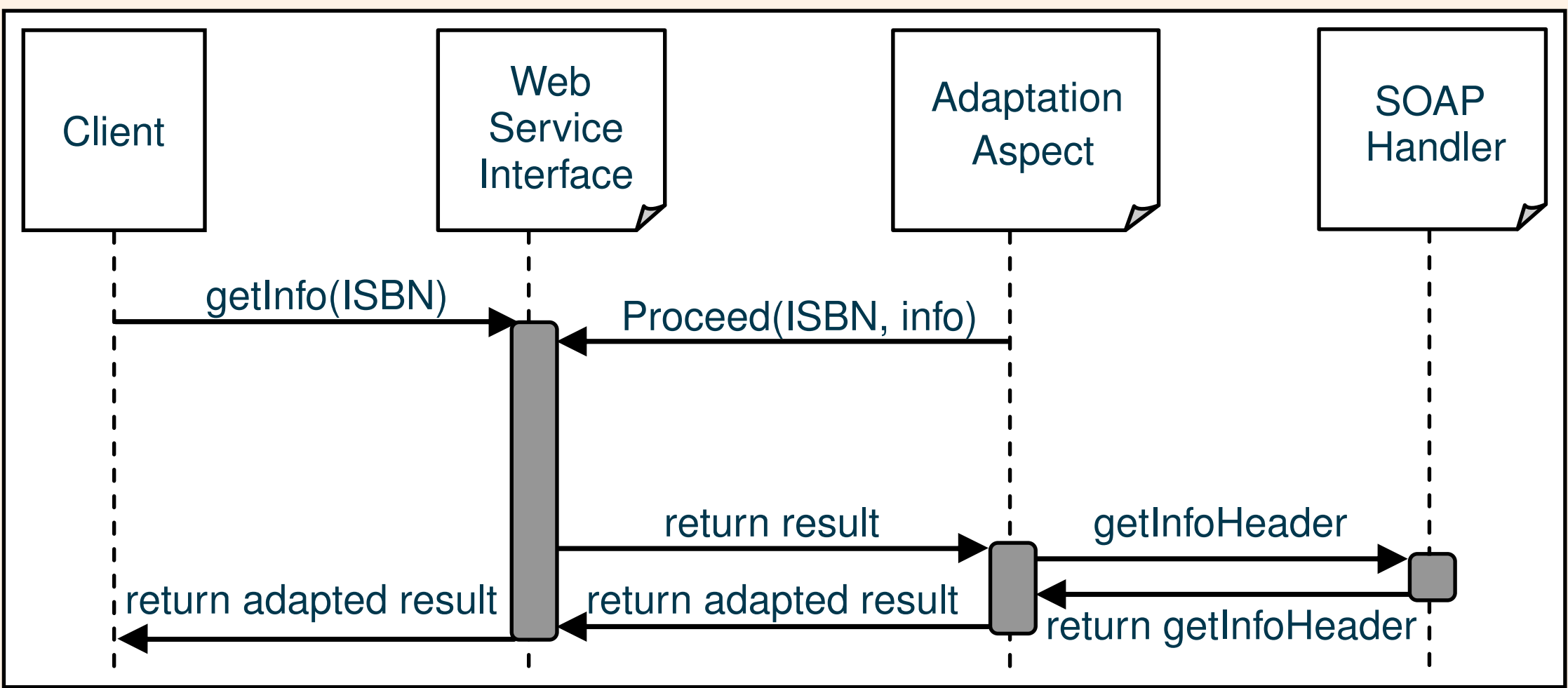
Mutación evolutiva

La prueba de mutaciones realiza cambios (*mutaciones*) en un programa para ver si sus pruebas pueden detectarlos. Proponemos la *mutación evolutiva*, en que se mejoran los casos de prueba de forma automatizada usando algoritmos genéticos para localizar mutantes no detectados y generar nuevos casos que los detecten. Nuestra herramienta **GAmera** implementa los algoritmos genéticos, usando los operadores de mutación para WS-BPEL 2.0 de **MuBPEL**.



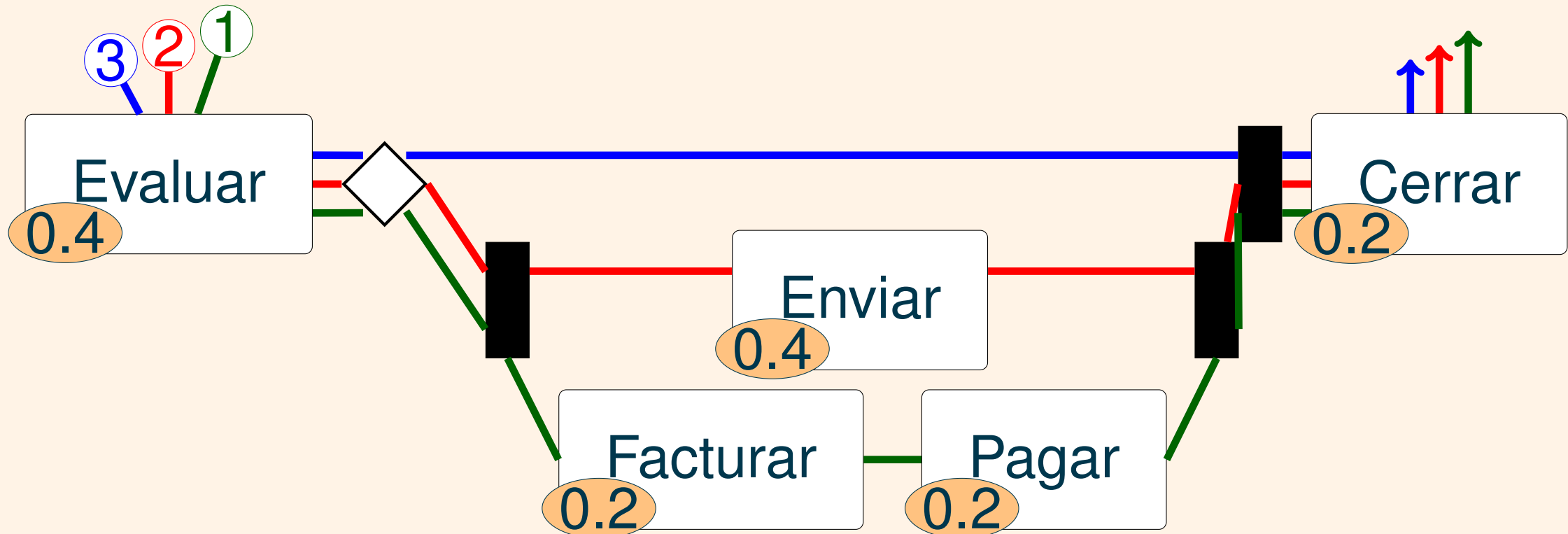
Adaptación de servicios al contexto

Actualmente las propuestas para adaptar los servicios web al contexto se centran en la adaptación del cliente. Nosotros proponemos no sobrecargar los clientes llevando a cabo la adaptación de forma no intrusiva en el servicio mediante un desarrollo dirigido por modelos y orientado a aspectos.



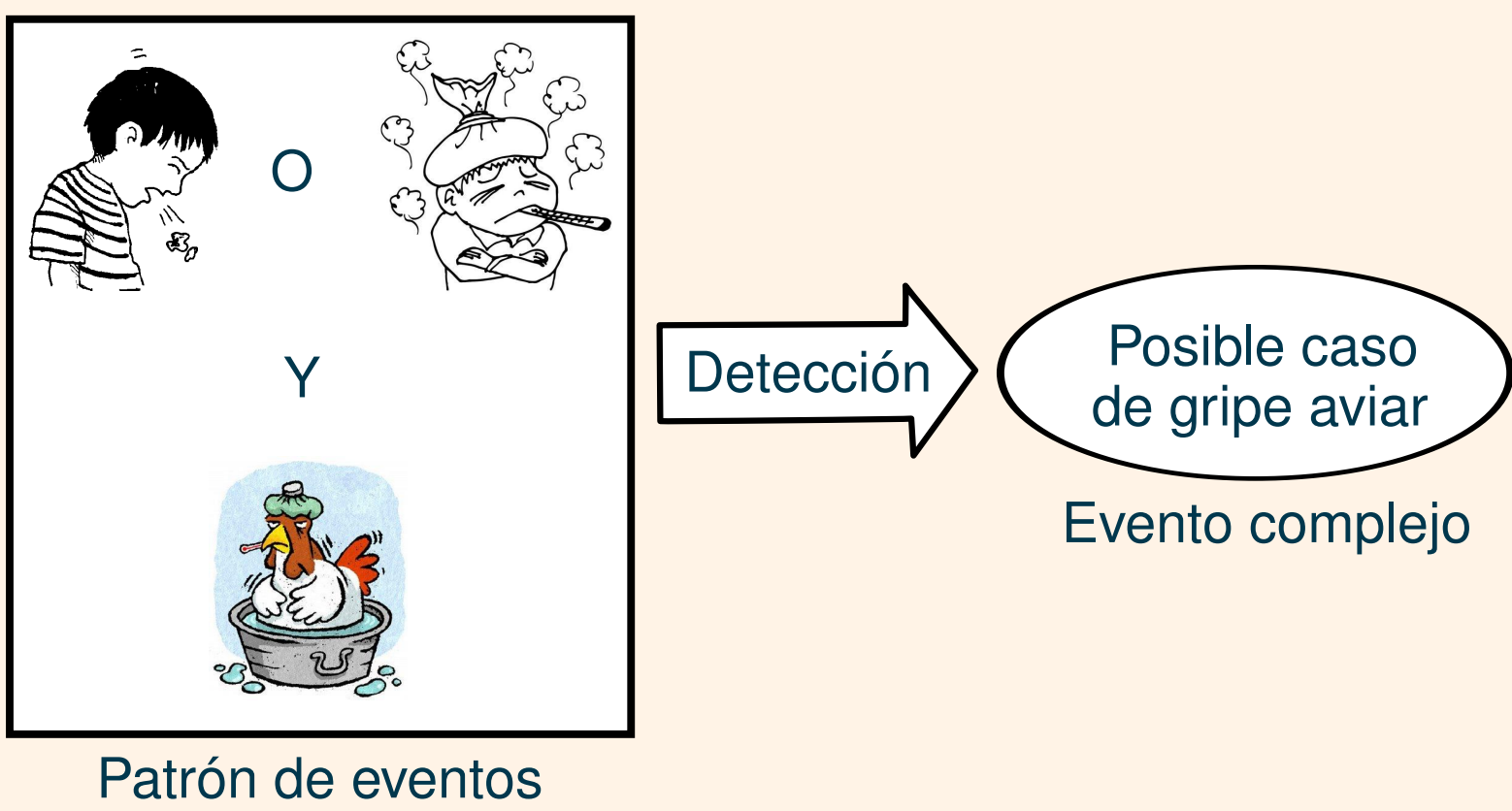
Pruebas dirigidas por modelos

Para obtener mejor software, se deberían realizar pruebas incluso antes de tener todo el código, usando modelos en su lugar. Dentro de la metodología **SODM+T**, hemos definido varios algoritmos de inferencia de requisitos de rendimiento en modelos de composiciones de servicios web. Estos requisitos se usarán después para generar casos de prueba concretos.



Procesamiento de eventos complejos

Mediante unos patrones de eventos, se pueden inferir nuevos eventos con mayor contenido semántico. Estos eventos complejos pueden tratarse en tiempo real, agilizando la toma de decisiones.



Verificación formal de software

Usamos ACL2, un dialecto de LISP diseñado para verificación. Es también una lógica computacional y un sistema de razonamiento automatizado que ayuda a demostrar propiedades de programas.

```
(defun mergesort (x)
  (if (or (atom x) (atom (rest x)) (< (len x) 16))
      (insertion-sort x)
      (let ((s (split x)))
        (merging (mergesort (first s))
                  (mergesort (second s))))))
(defthm sortedp-mergesort (sortedp (mergesort x)))
```

Generación dinámica de invariantes (con SPI&FM)

Colaboramos con miembros del grupo SPI&FM (UCA) en generar propiedades de ciertos puntos de una composición WS-BPEL a partir de trazas de ejecución, mediante la herramienta **Takuan**.

