



# Lenguajes Visuales Orientados a Aspectos

Fernando Asteasuain – fasteasuain@dc.uba.ar - LAFHIS

Departamento de Computación, FCEyN, Universidad de Buenos Aires

## ¿Qué es la Orientación a Aspectos?

Mecanismo avanzado de modularización.  
Aspecto = **pointcut** (dónde) + **advice** (qué)

Ejemplo: Luces Interiores



**Aspecto Luz Interior**  
**pointcut** : "Al abrirse una puerta"  
**advice** : "Prender luces Interiores"

## Interferencia entre Aspectos

Dos aspectos **interfieren** cuando se aplican en el mismo lugar de interés



Manejo Interferencia == detectar y determinar orden correcto de aplicación

**Se requieren notaciones formales fáciles de entender, razonar y manipular.**

## Ejemplo de Interferencia

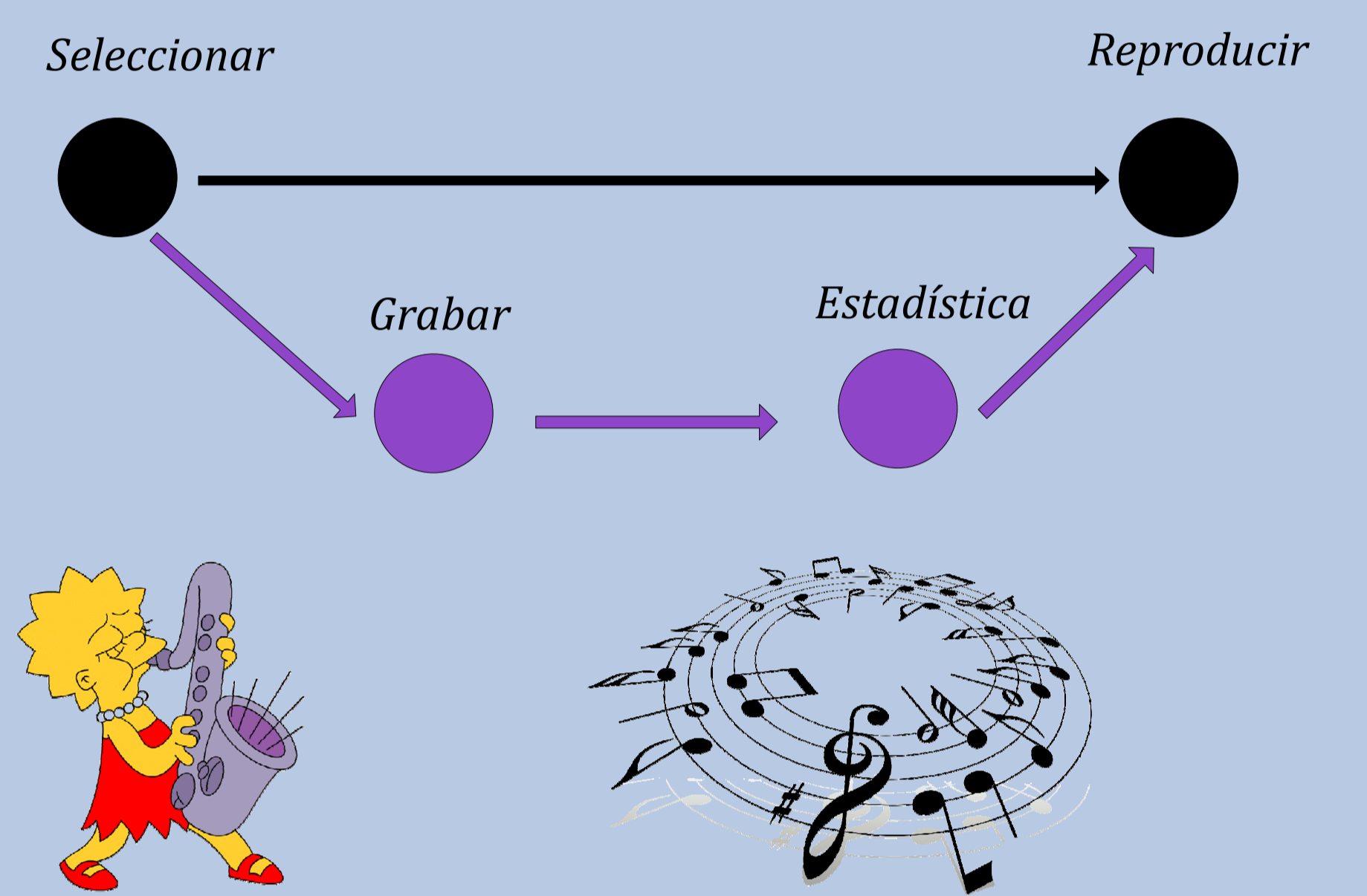
Sistema "Rockola" que reproduce música +  
Aspecto 1: estadísticas de la canción  
Aspecto 2: guardar últimas canciones



¿Qué aspecto debe aplicarse primero? ¿Importa? ¿Porqué?

## Ejemplo Rockola

Una especificación inicial podría ser



## Notaciones Declarativas: FVS

**FVS (Featherweight Visual Scenarios)** lenguaje declarativo basado en escenarios

### Reglas FVS

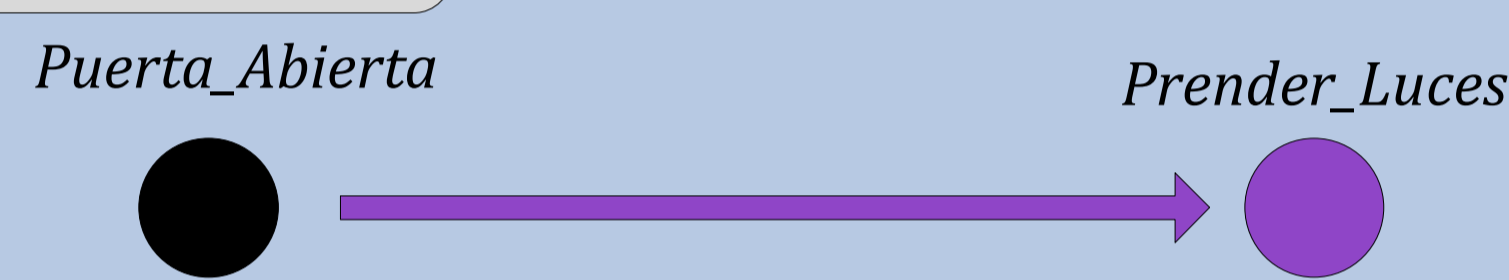
un antecedente (dibujado en negro) y uno o más consecuentes (color violeta)

**Semántica:** Siempre que el sistema cumple el antecedente, debe cumplir también al menos uno de los consecuentes.

## FVS Orientado a Aspectos

**Antecedente == pointcut**  
**Consecuente == advice**

Ejemplo Luces

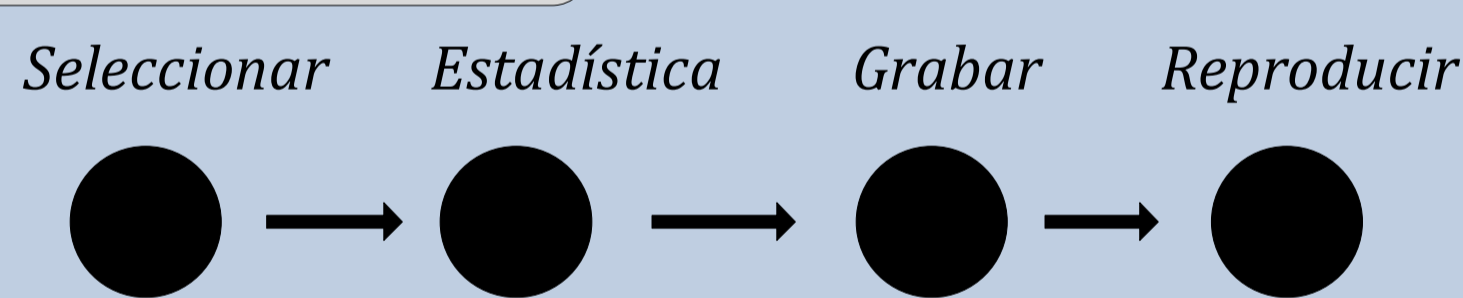


Siempre que ocurre el evento "Puerta\_Abierta" (antecedente) debe ocurrir el evento "Prender\_Luces" (consecuente)

## Anti-Escenarios

Representa una violación de una regla

Ejemplo Rockola

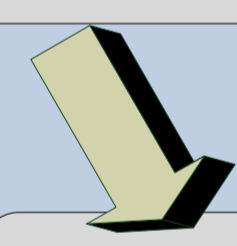


Si las estadísticas se hacen primero es un comportamiento no válido

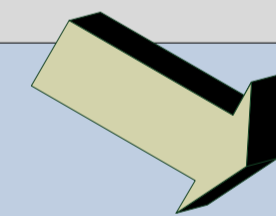
Generados automáticamente en FVS

## Analizando Anti-Escenarios

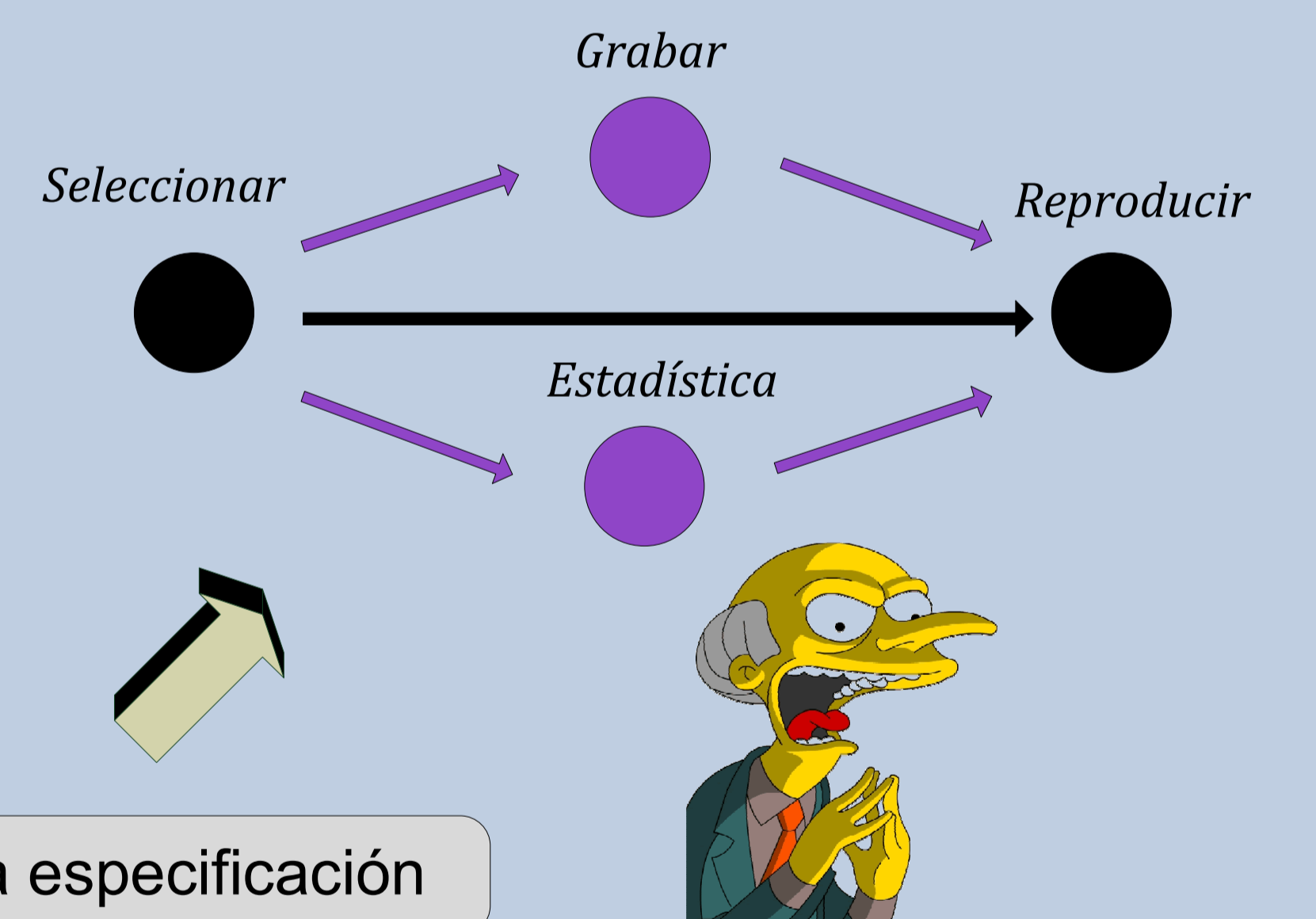
Al analizar el anti-escenario el usuario podría darse cuenta que el orden de los aspectos no es importante: **no importa qué va primero, sino que ambos sucedan**



La especificación inicial era muy restrictiva: dejaba afuera comportamiento válido



Se obtiene una nueva especificación



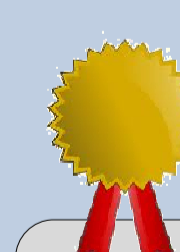
## Aportes de FVS



**Formalismo adecuado para entender, razonar y manipular especificaciones orientadas a aspectos.**



**Analizar anti-escenarios permite entender y resolver la interferencia entre aspectos**



Más Features

**Declaratividad, y Razonamiento a través de trazas válidas**

### Papers

- (1) F. Asteasuain, V. Braberman "Exploring visual scenarios as an aspect-oriented modeling language" ASSE 2010
- (2) F. Asteasuain, V. Braberman "FVS: A declarative aspect oriented modeling language" EJS 2011
- (3) F. Asteasuain, V. Braberman "Visual Scenarios for addressing the Aspect Interference Problem" ASSE 2011

### TRABAJO FUTURO



- Desarrollo Herramienta Visual basada en FVS.
- Aplicar FVS Orientado a Aspectos en casos de estudios interesantes.

- Posible Tema de Licenciatura