

Un ejemplo de sistema: Resolución de Sudokus en grupo

Ingeniería del Software II

Descripción general

- Se trata de un sistema cliente-servidor
- Los clientes se conectan al servidor, seleccionan un Sudoku de una lista y comienzan a resolverlo
- Cada vez que uno pone un número, éste aparece en todos los clientes que están resolviendo el mismo sudoku

Sudokus

Servidor: Puerto:

11

6							
		9			1		
1	8			5	4	2	
	1		4			6	
2	4						3
		8			6		1
		3	1	2			
7			9			3	

Recibo cambio sobre el 11
 Recibo cambio sobre el 11
 Recibo cambio sobre el 11
 Recibo cambio sobre el 11

Sudokus

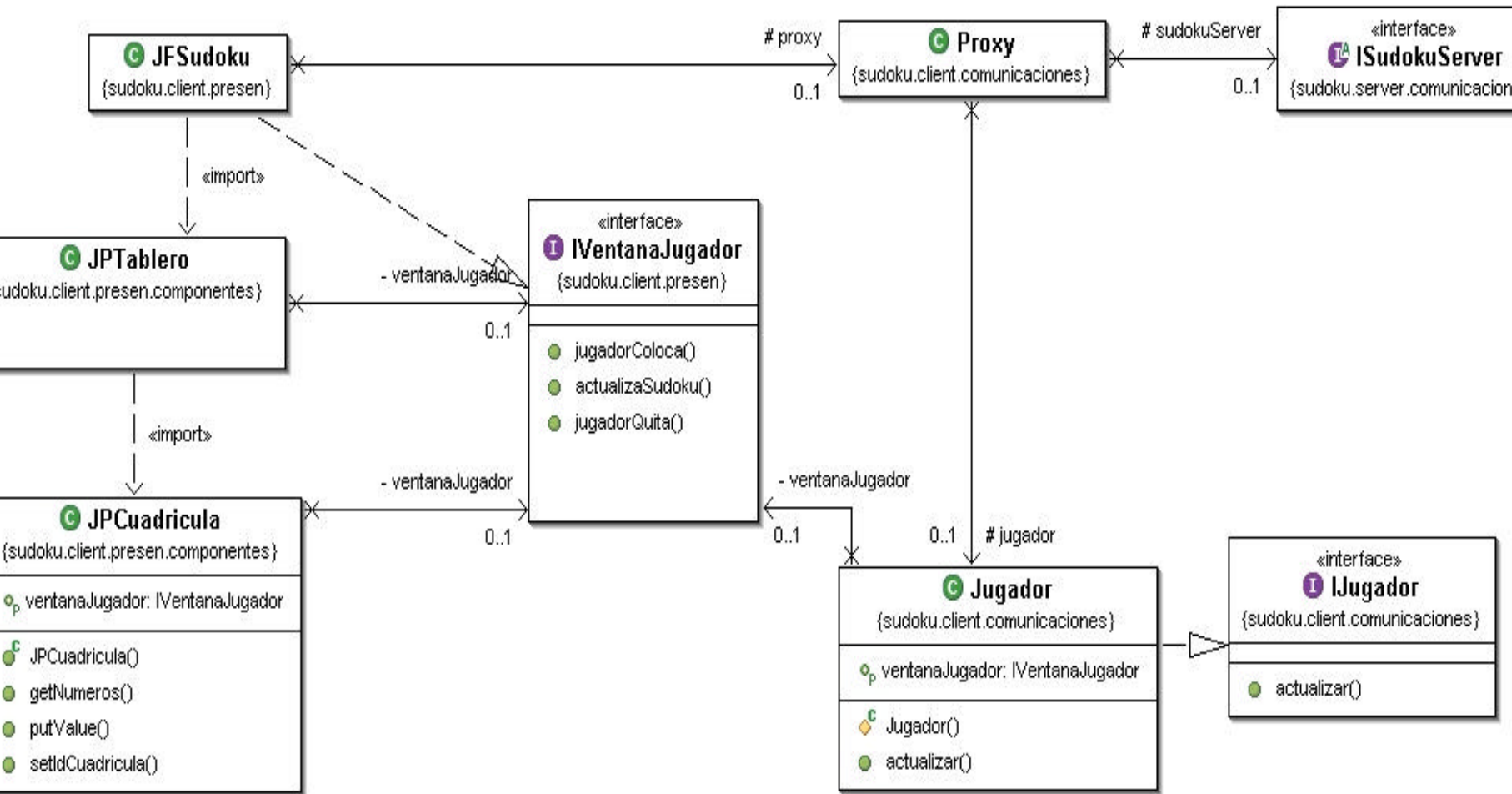
Servidor: Puerto:

11

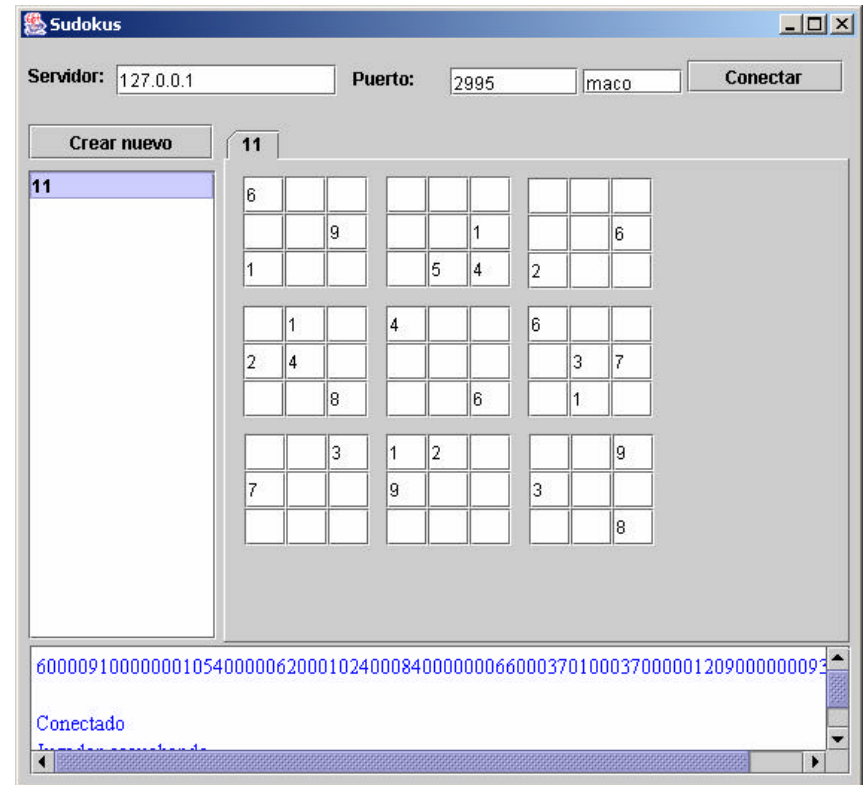
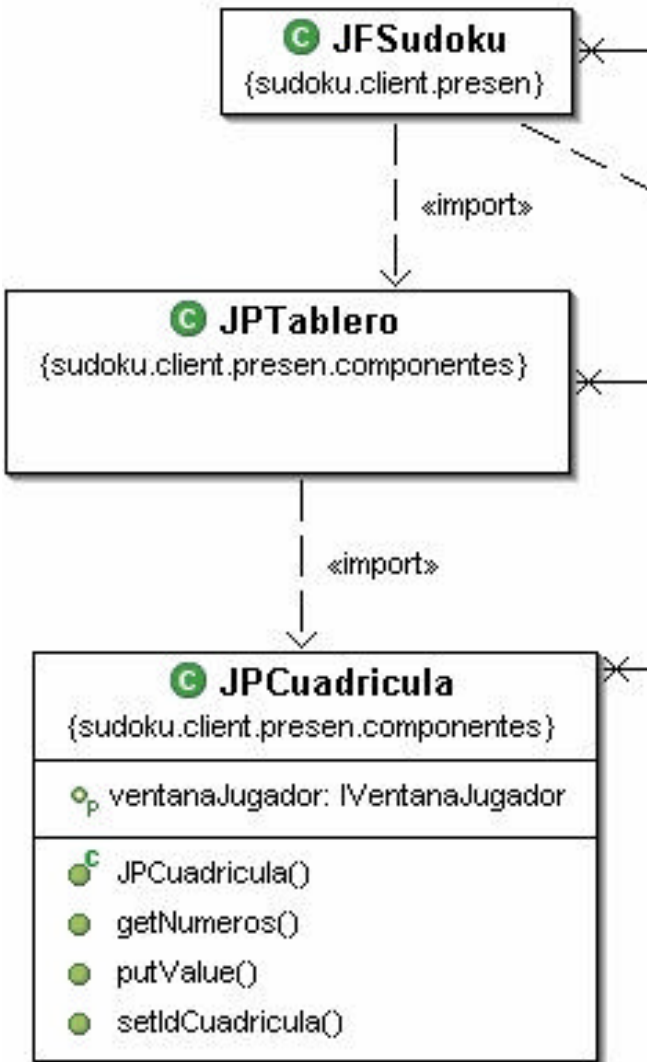
6							
		9			1		
1	8			5	4	2	
	1		4			6	
2	4						3
		8			6		1
		3	1	2			
7			9			3	

Recibo cambio sobre el 11
 Recibo cambio sobre el 11
 Recibo cambio sobre el 11
 Recibo cambio sobre el 11

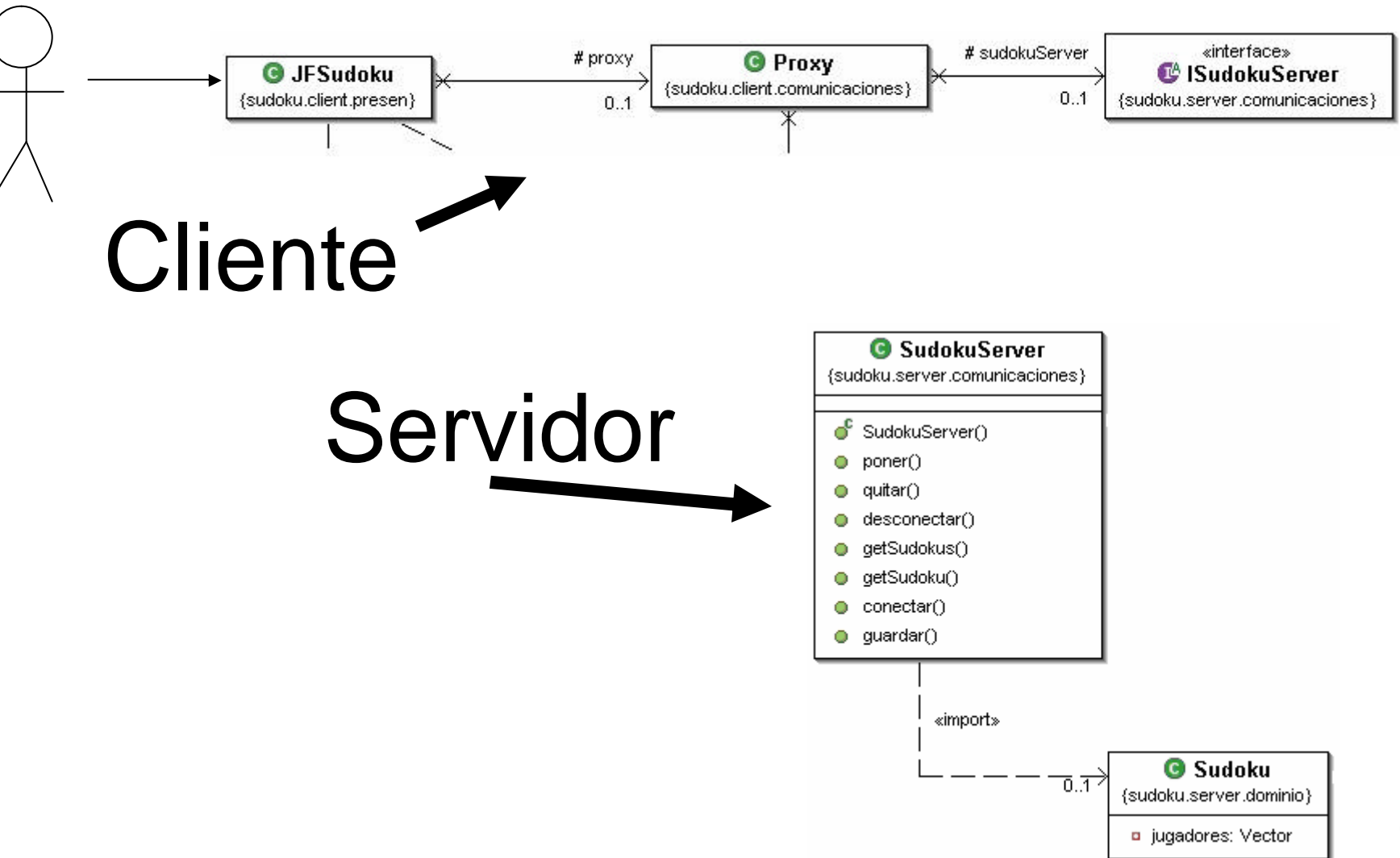
Estructura del cliente



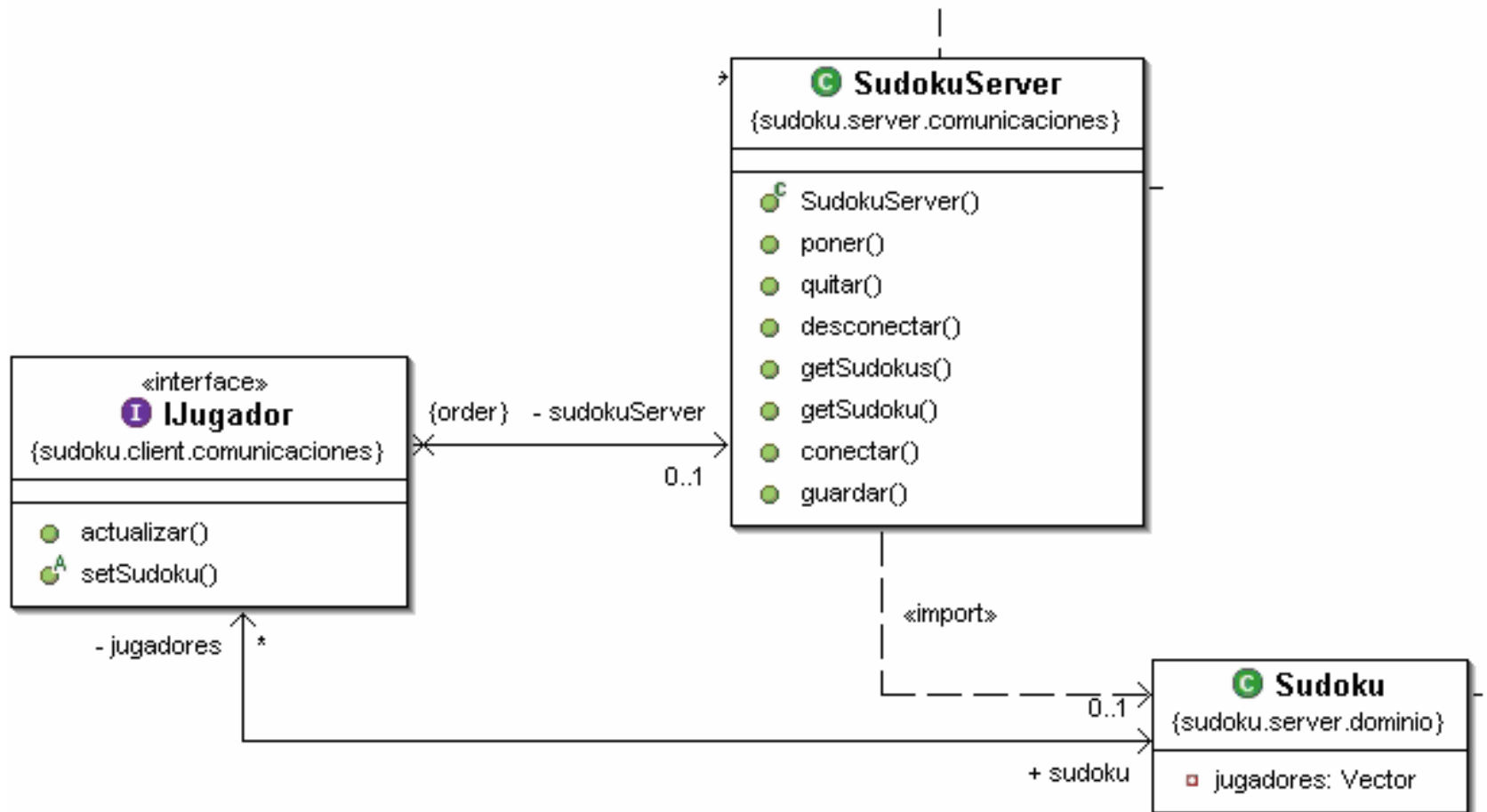
Capa de presentación



Envío de mensajes al servidor (poner un número)

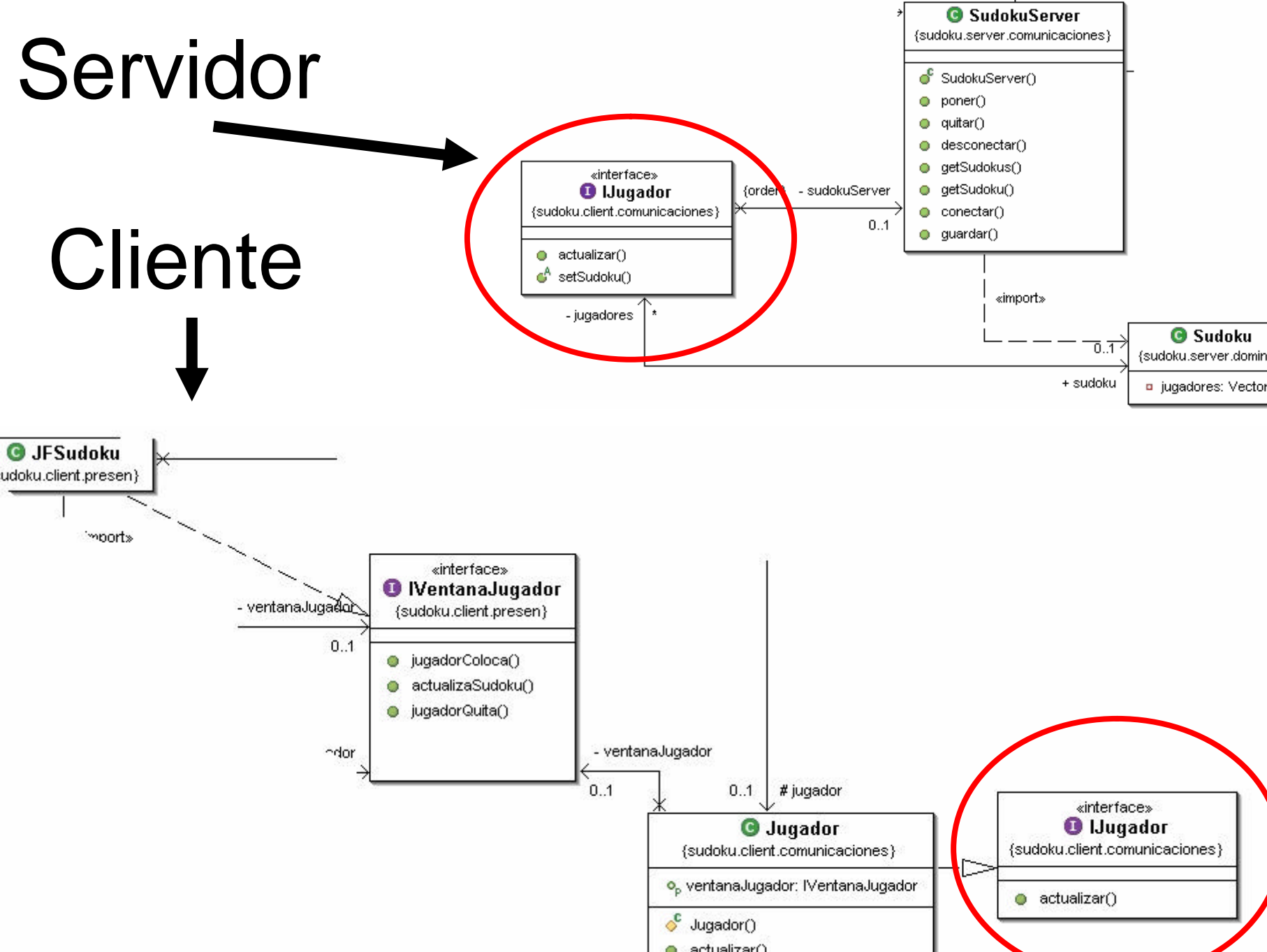


Procesamiento del mensaje en el servidor

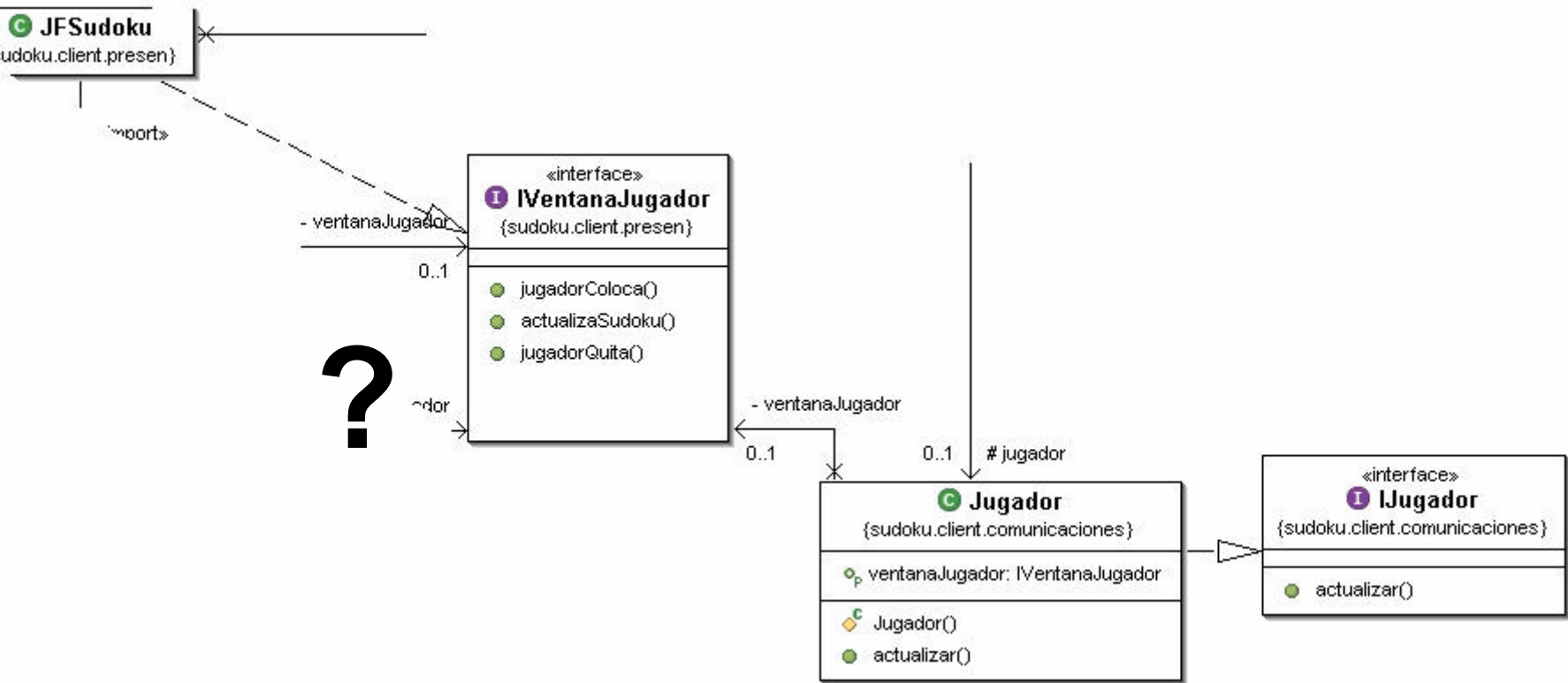


Servidor

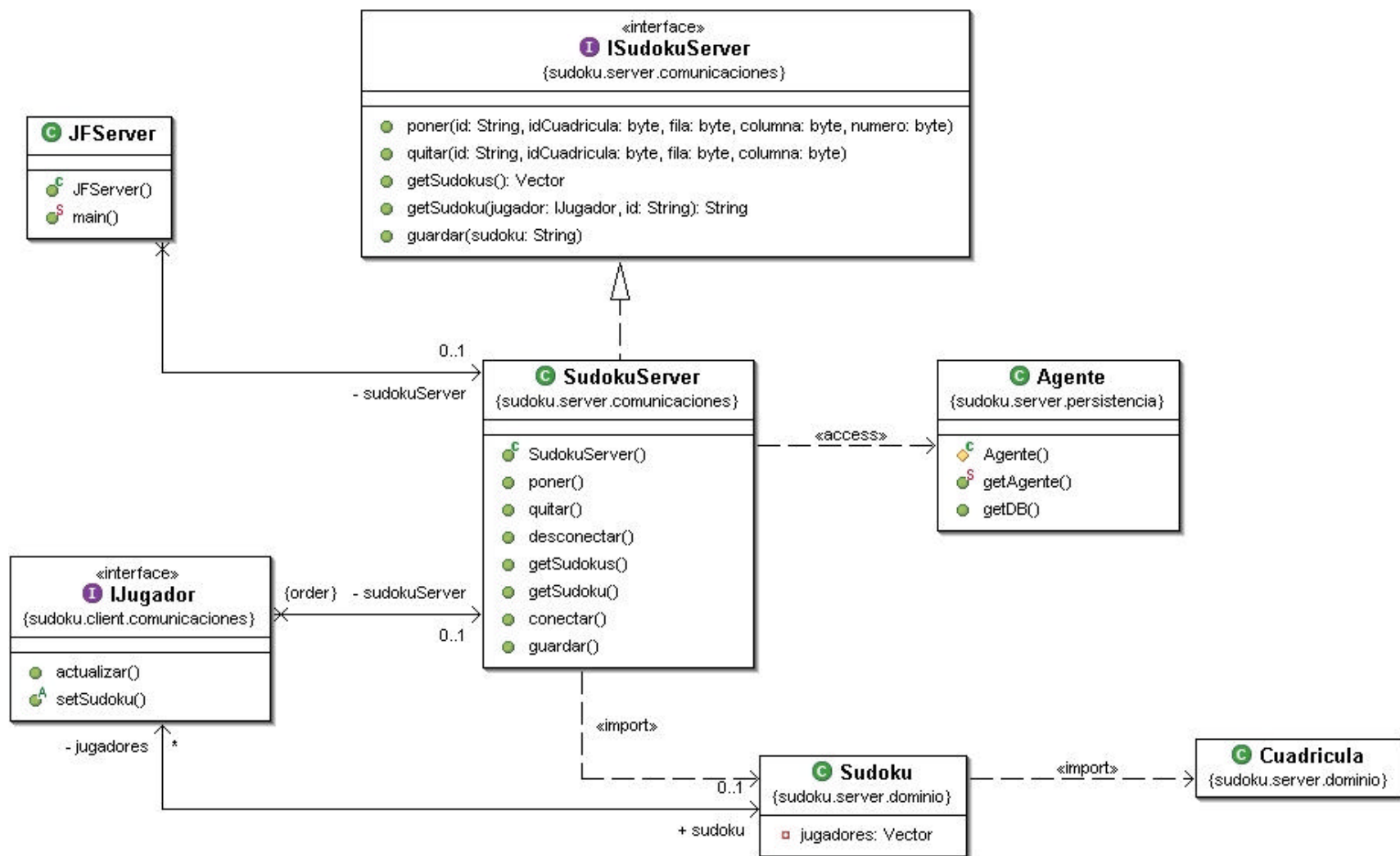
Cliente



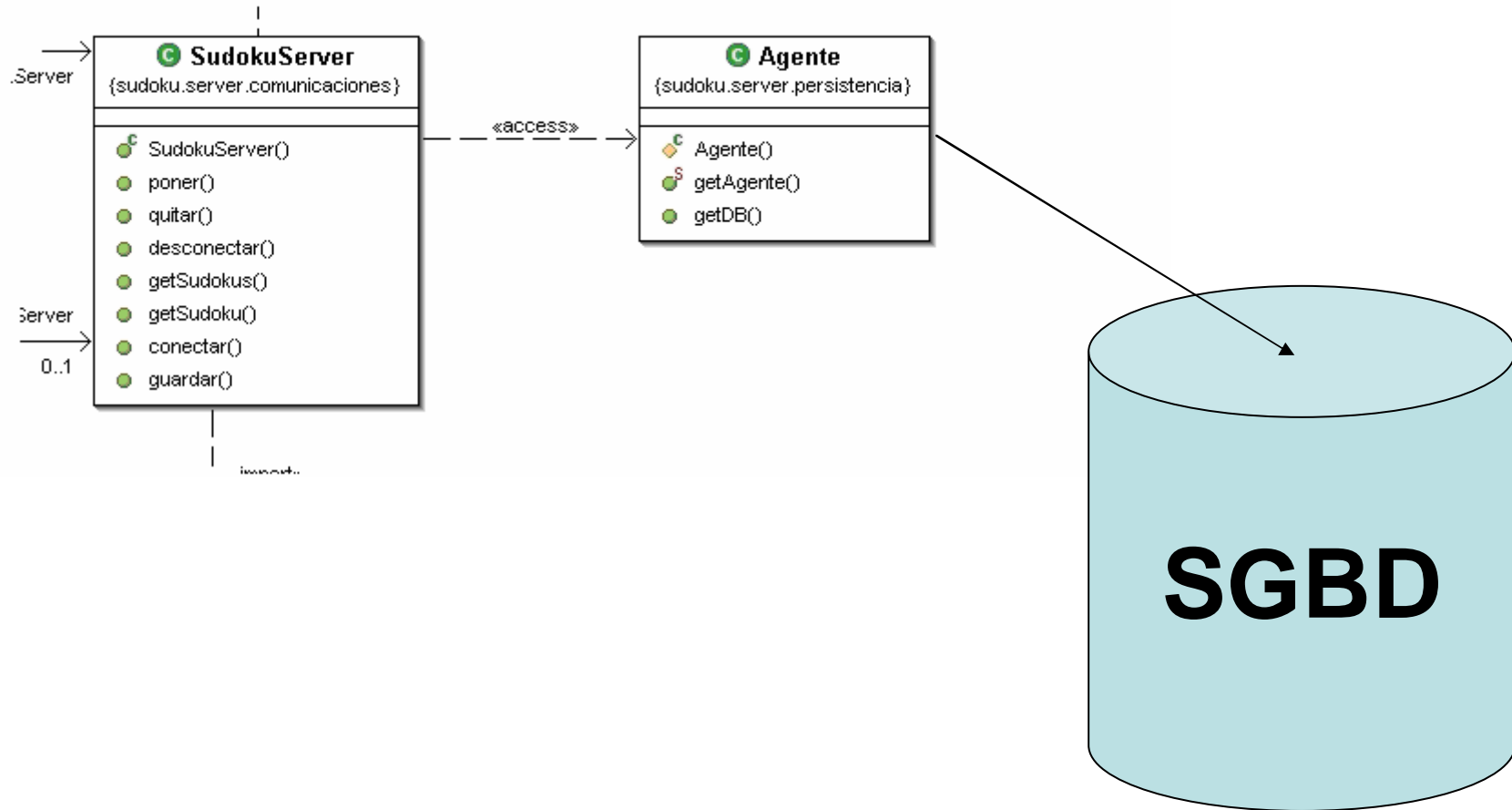
¿Por qué hay una interfaz *IVentanaJugador* en el cliente?



Estructura del servidor



Acceso a la base de datos (en el servidor)



Resumiendo (I)

- Tenemos dos aplicaciones sencillas que se comunican mediante *rmi*
- Ambas tienen estructura multicapa
 - Cliente: presentación, dominio, comunicaciones
 - Servidor: presentación, dominio, comunicaciones, persistencia

Resumiendo (II)

- Diferentes patrones (soluciones buenas a problemas frecuentes)
 - Separación en capas
 - Separación modelo-vista (observador)
 - Agente de base de datos
 - Singleton (en el mismo agente)
 - Alta cohesión (cada clase hace exactamente lo que tiene que hacer, y nada más)
 - Bajo acoplamiento (cada clase conoce a quien debe conocer, y a nadie más)