

Programación orientada a objetos

PRÁCTICA OBLIGATORIA

Simulador: entidades implicadas

- **Banco:**
 - posee una lista de clientes que desean invertir en bolsa.
- **Bróker:**
 - tiene capacidad de realizar operaciones con la bolsa.
- **Bolsa:**
 - gestiona el precio de las acciones de un listado de empresas.

Simulador: comunicación entre entidades

- La comunicación entre **bróker** y **bolsa** se realiza mediante:

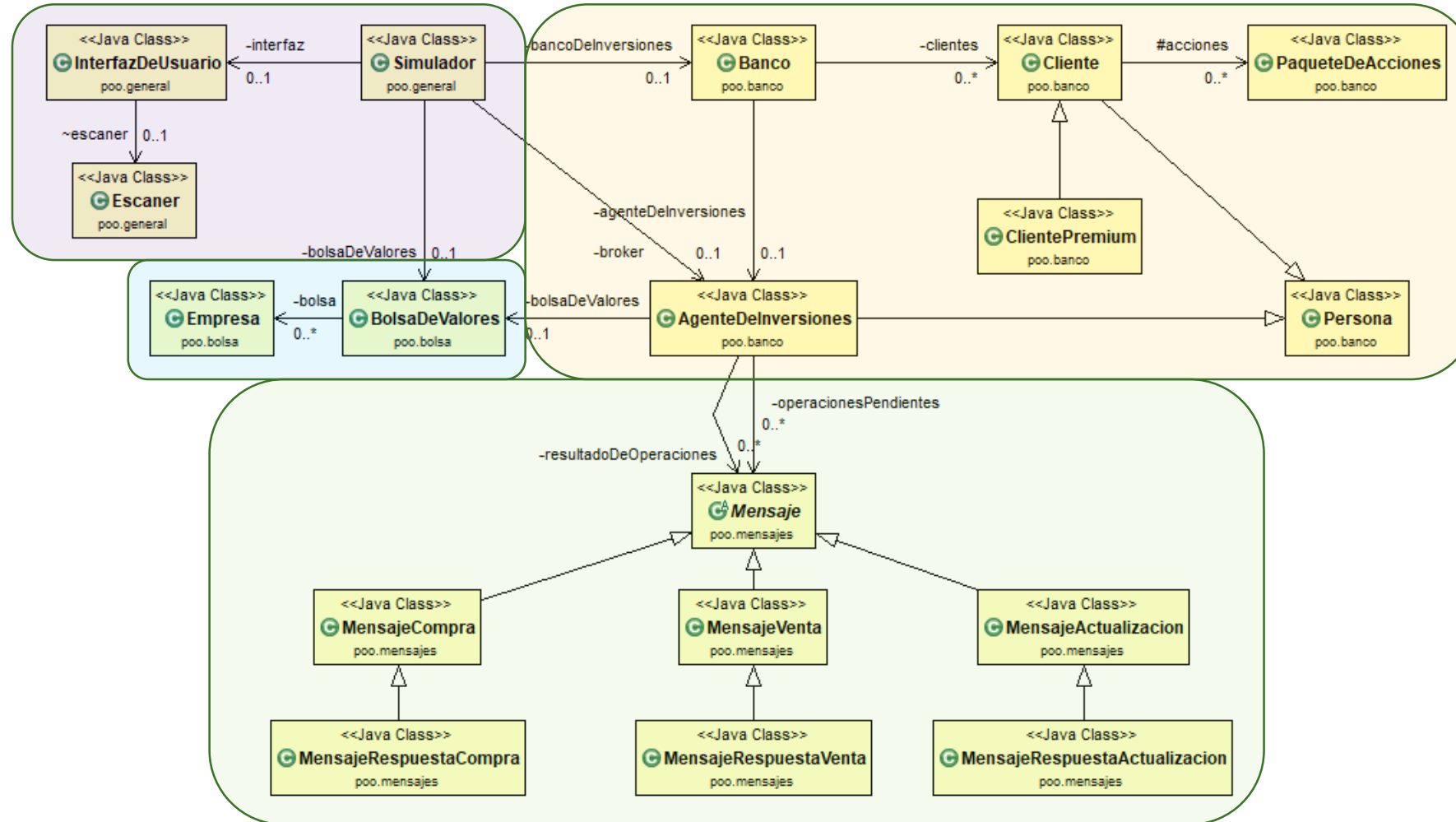
public String intentaOperacion(String mensajeCodificado) {

- El **banco** construye los mensajes en función de las necesidades de los clientes.
- El **bróker** codifica los mensajes para enviárselos a la **bolsa**.
- La **bolsa** decodifica los mensajes, realiza la operación y devuelve una respuesta codificada al **bróker**.
- El **banco** actualiza la información de los clientes en función de la respuesta de la **bolsa**.

Simulador: ejemplo de codificación de mensajes

- Campos de la solicitud de compra:
 - Identificador: 5052
 - Nombre de cliente: John Nash
 - Nombre de la empresa: Tesla
 - Inversión máxima: 3000 €
- Texto codificado enviado a la bolsa por el bróker:
`5052|John Nash|Tesla|0003000,00`
- Respuesta de la bolsa después de realizar la operación:
`5052|John Nash|true|00000002|0001033,00|0000934,00`

Simulador: uml



Simulador: ejecución

```
0.- Salir
----- ESTADO -----
1.- Imprimir estado de los clientes
2.- Imprimir estado de la bolsa
----- BANCO -----
3.- Añadir cliente
4.- Eliminar cliente
5.- Realizar copia de seguridad
6.- Restaurar copia de seguridad
7.- Mejorar cliente a premium
8.- Solicita recomendación de inversión
----- BOLSA -----
9.- Añadir empresa a la bolsa
10.- Eliminar empresa de la bolsa
11.- Actualización de valores
12.- Realizar copia de seguridad
13.- Restaurar copia de seguridad
----- OPERACIONES -----
14.- Solicitar compra de acciones
15.- Solicitar venta de acciones
16.- Solicitar actualización de valores
----- BRÓKER -----
17.- Imprimir operaciones pendientes
18.- Ejecutar operaciones pendientes
-----
```

- El operador puede ejecutar cualquier opción.
- El programa debe contemplar acciones no disponibles.
- Si una acción requiere información adicional, el programa los solicitará por pantalla.
- El programa debe comprobar los datos introducidos antes de llevar a cabo la operación.
- Se deben imprimir todos los mensajes necesarios para demostrar la ejecución de cada acción.
- Después de ejecutar cada acción, se vuelve a mostrar el menú general.

...

