



## Problema Ruleta online (3 puntos)

Una famosa empresa de la costa oeste Californiana le ha contratado para trabajar en un prototipo de su próximo casino online, que está siendo implementado con tecnologías J2EE. Más en detalle, le han pedido que participe activamente en el desarrollo de una aplicación denominada ruleta electrónica (*ruletaJunio08.ear*).

Dicha ruleta consta de, tal y como muestra la figura 1, tres componentes primarios: uno denominado *Ruleta*, otro *BolsaCliente* y otro de apuestas *Apuesta* (los tres son beans de entidad de tipo CMP). Dichos componentes ofrecen entidades de negocio que permiten almacenar la lógica necesaria para hacer girar la ruleta, controlar el dinero que posee cada usuario y realizar apuestas. Finalmente hay un último componente, *TestApuestas*, que es el encargado de realizar ciertas pruebas unitarias sobre dicho sistema. Este último componente es un bean de sesión con estado (*statefull*) que usted tiene que desarrollar.

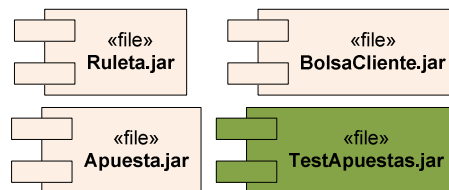


Figura 1: ruletaJunio08.ear

De este sistema, sabe lo siguiente:

- El componente *Ruleta* (cuya interfaz `softcom.RuletaHome` está disponible en `ejb/Ruleta`) ofrece los siguientes métodos remotos:
  - `RuletaRemote create(String id) throws CreateException, RemoteException;` crea una nueva ruleta
  - `void remove() throws RemoveException, RemoteException;`
  - `int lanzaBola() throws RemoteException;` devuelve un número entre 0 y 36, correspondiente al resultado de un lanzamiento de la bola de la ruleta.
- El componente *BolsaCliente* (cuya interfaz `softcom.BolsaClienteHome` está disponible en `ejb/BolsaCliente`) ofrece los siguiente métodos:
  - `BolsaRemote create(int saldoInicial) throws CreateException, RemoteException;`
  - `void remove() throws RemoveException, RemoteException;`
  - `int getAvailable();` dice el saldo que queda en la bolsa

- `ApuestaRemote generaApuestaPleno(int cantidad, int numero) throws RemoteException;` genera una apuesta a un número (`numero`) por una cantidad determinada (`cantidad`).
- `void cobroApuesta(ApuestaRemote apuesta) throws RemoteException;`
- El componente `Apuesta` (cuya interfaz `ApuestaHome` está disponible en `ejb/Apuesta`) ofrece los siguiente métodos:
  - `void validate(int bolaQueSalioEnLaRuleta) throws RemoteException;`

Y se le pide que:

1. (1 punto) Diseñe la vista cliente (*interfaces home y remote*) del componente `TestApuestas` (cuya interfaz `softcom.TestApuestaHome` estará disponible en `ejb/TestApuestas`) de tal manera que el componente resultante ofrezca los siguientes métodos remotos:
  - `void setUp()`: crea una ruleta general, 2 bolsas de dinero (con 1000 euros cada una), y 36 apuestas (de 1 euro cada una) a cada uno de los 36 números de la ruleta, creadas a partir del dinero del primer apostante.
  - `void tearDown()`: destruye la ruleta, las dos bolsas de dinero y las 36 apuestas creadas.
  - `void testPleno()`: Lanza la bola de la ruleta hasta que el resultado obtenido sea distinto de cero, valida todas las apuestas generadas (llamando a su método `validate()`), y por último hace que cada cliente cobre sus apuestas.  
 En caso de que el saldo de la bolsa 1, tras haber cobrado las apuestas, no sea de 999 euros lanza una excepción remota con el siguiente mensaje: "[testPleno] Error: la validación de la apuesta de pleno es incorrecta".
2. (2 puntos) Codifique la clase `TestApuestasBean`, de tal manera que el componente resultante sea un *bean de sesión con estado*.