

Ingeniería del Software II. Curso 2005/2006.

Enunciado de la práctica del segundo semestre.

Para este segundo semestre es preciso ampliar la práctica del primero de varias formas, que se describen a continuación.

1. Fidelización.

En el momento de realizar el pago, el empleado preguntará al cliente que si dispone de la tarjeta *Fidel*, que acumula puntos en una cuenta del cliente. En caso afirmativo, se comunicará el importe de la compra utilizando y el número de la tarjeta *Fidel* a un servicio *rmi* llamado *IFidel* ofrecido en *rmi://161.67.38.10:2006/fidel*. La signatura de la operación es la siguiente:

`+fidel(numeroDeTarjeta:long, importe:double, dniAlumno:String, pwd:String)`

...donde *dniAlumno* y *pwd* se corresponden con el login y la contraseña utilizados para acceder a la intranet de la asignatura.

En el anexo se muestra el código de un cliente de prueba.

Los archivos necesarios para hacer la conexión al servidor *rmi* se encuentran en la página web de la asignatura, empaquetados en *stubs.jar*.

2. Lenguaje de comandos.

Debe integrarse un pequeño servidor *rmi* en el sistema para que, desde remoto, se puedan realizar distintas operaciones procesando el siguiente lenguaje de comandos, usando obligatoriamente el patrón

Intérprete:

Comando = CrearOferta | CrearProducto | CambiarPrecios | Consultar

CrearOferta = CrearNPorM | CrearNMásM

CrearNPorM = 'crear' código_producto n 'x' m

CrearNMásM = 'crear' n '(' código_producto ')' '+' m '(' código_producto ')'

CambiarPrecios = 'cambiar' código_producto ['+' | '-'] porcentaje '%'

Consultar = 'ofertas' | 'productos'

Ejemplos:

crear 7 3x2 ← crea una oferta de 3x2 para el producto 7

crear 2(7) + 1(3) ← crea una oferta 2+1: compras 2 uds. del prod. 7 y te llevas una del 3

cambiar 7 +3.5% ← sube el precio del producto 7 un 3.5%

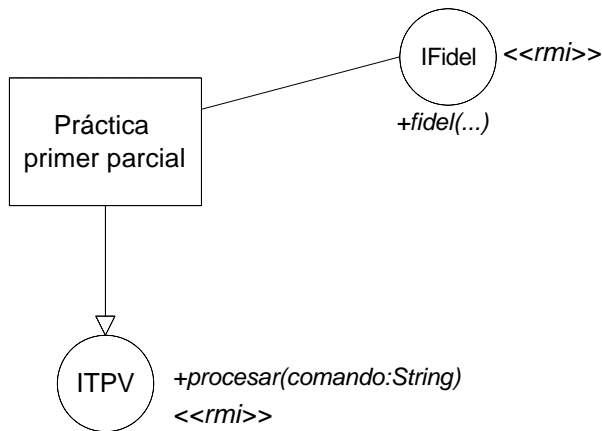
cambiar 7 3.5% ← sube el precio del producto 7 un 3.5%

cambiar 7 -3.5% ← baja el precio del producto 7 un 3.5%

ofertas ← devuelve una cadena con la lista de ofertas

productos ← devuelve una cadena con la lista de productos

El servicio de ejecución de comandos debe implementarse como un servidor rmi llamado *ITPV* que ofrezca la operación *procesar(comando:String)*. Será preciso construir un pequeño cliente que permita hacer en el laboratorio una demostración del funcionamiento del servicio.



Anexo.

```

package clientefidel;

import java.net.MalformedURLException;
import java.rmi.Naming;
import java.rmi.NotBoundException;
import java.rmi.RemoteException;
import java.sql.SQLException;
import dominio.IFidel;

public class Prueba {
    public static void main(String[] args) {
        try {
            IFidel f=(IFidel) Naming.lookup("rmi://localhost:2006/fidel");
            try {
                f.fidel(5, 100, "1000", "1000");
            } catch (SQLException e1) {
                e1.printStackTrace();
            }
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (RemoteException e) {
            e.printStackTrace();
        } catch (NotBoundException e) {
            e.printStackTrace();
        }
    }
}
  
```