**INSTITUTO POLITÉCNICO NACIONAL**

**Escuela Superior de Cómputo (ESCOM)**

**PROFESOR**: Axel Ernesto Moreno Cervantes.

**Unidad de Aprendizaje**: Aplicaciones de Comunicaciones en Red.

**TRABAJO:** Practica 2.- Carrito de Compras.

**ALUMNO:**

* Ángel Miranda Pedro.
* Monroy Martos Elioth.

**GRUPO:** 3CM7

# Introducción

Un socket es una forma mediante la cual dos programas pueden intercambiar cualquier flujo de datos. Un socket comunmente es definido por un par de direcciones IP local y remota, un protocolo de transporte y un par de números de puerto local y remoto.

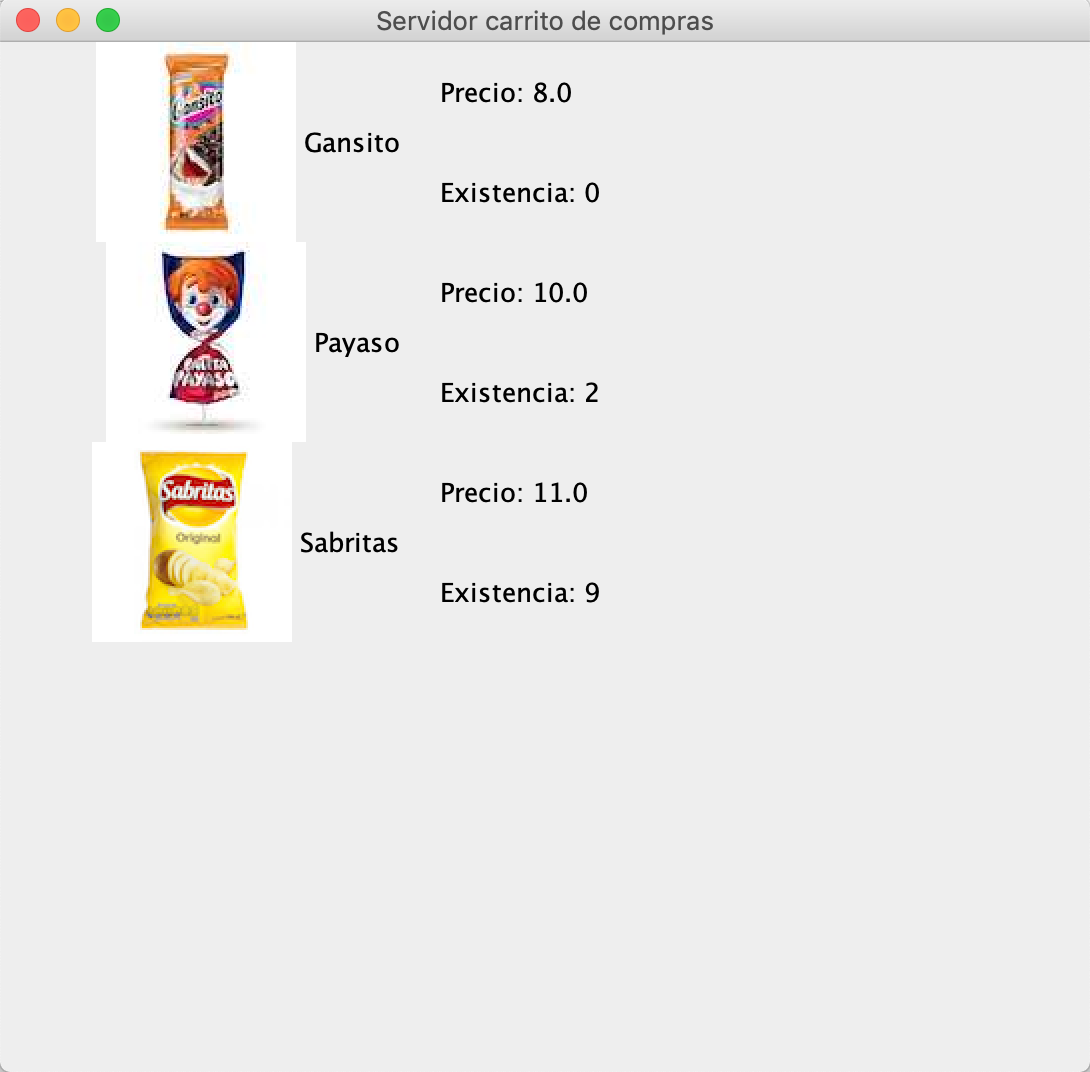
TCP: Este protocolo está orientado a la conexión. Cuando una máquina A envía datos a una máquina B, la máquina B es informada de la llegada de datos, y confirma su buena recepción. Aquí interviene el control CRC de datos que se basa en una ecuación matemática que permite verificar la integridad de los datos transmitidos. De este modo, si los datos recibidos son corruptos, el protocolo TCP permite que los destinatarios soliciten al emisor que vuelvan a enviar los datos corruptos.

El desarrollo de la presente práctica consiste en hacer una aplicación cliente-servidor que funcione como un carrito de compras, donde distintos clientes puedan comprar productos que un servidor muestra, a su vez, cada vez que un cliente compra un producto, la existencia del mismo se ve afectada.

# Desarrollo

La práctica se dividió en tres secciones principales, una es una interfaz que sirve como pantalla para “Administrador” desde donde se puede consultar los productos y la cantidad que se tiene en existencia de los mismos.

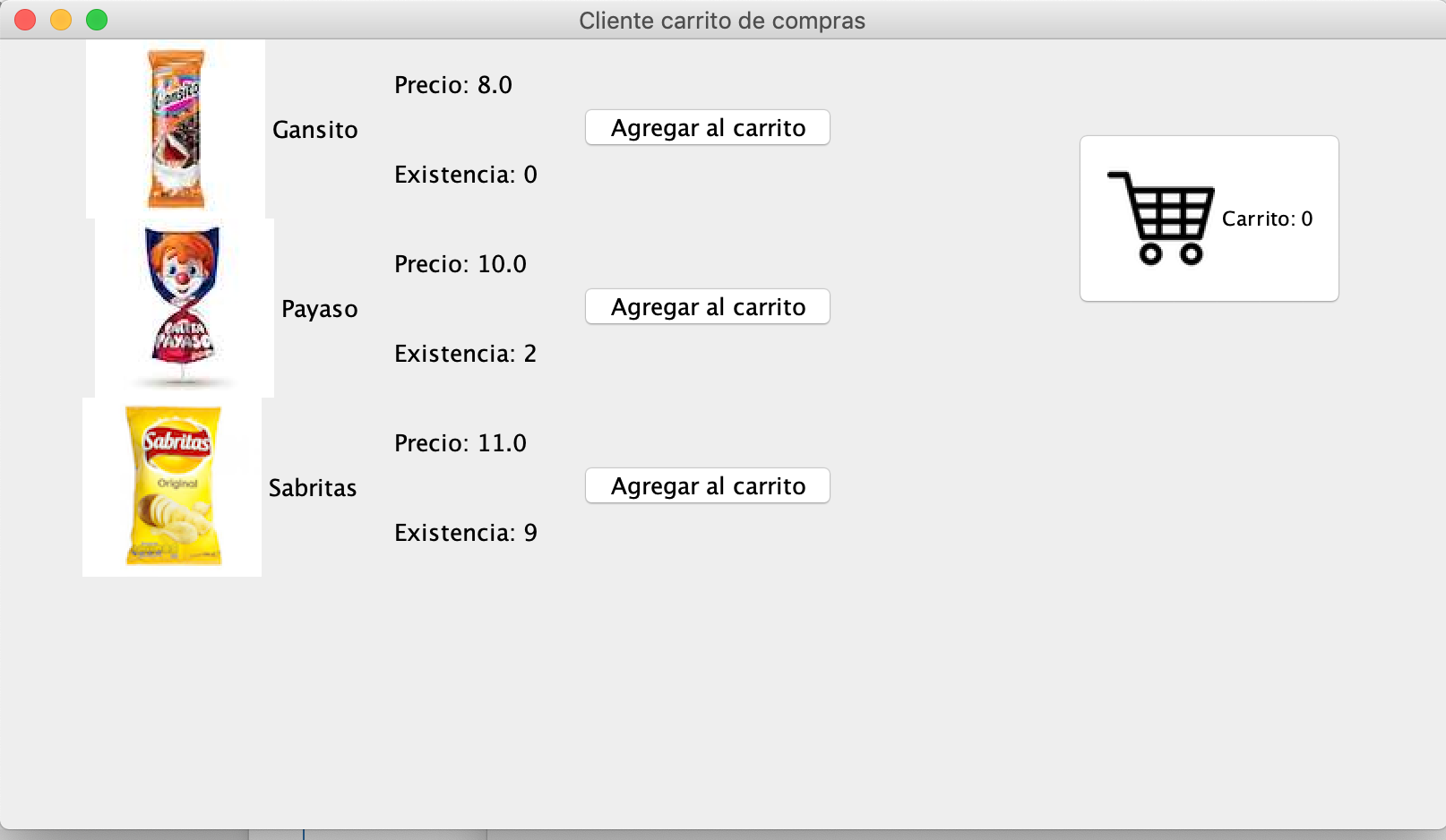
Servidor:



Cliente:

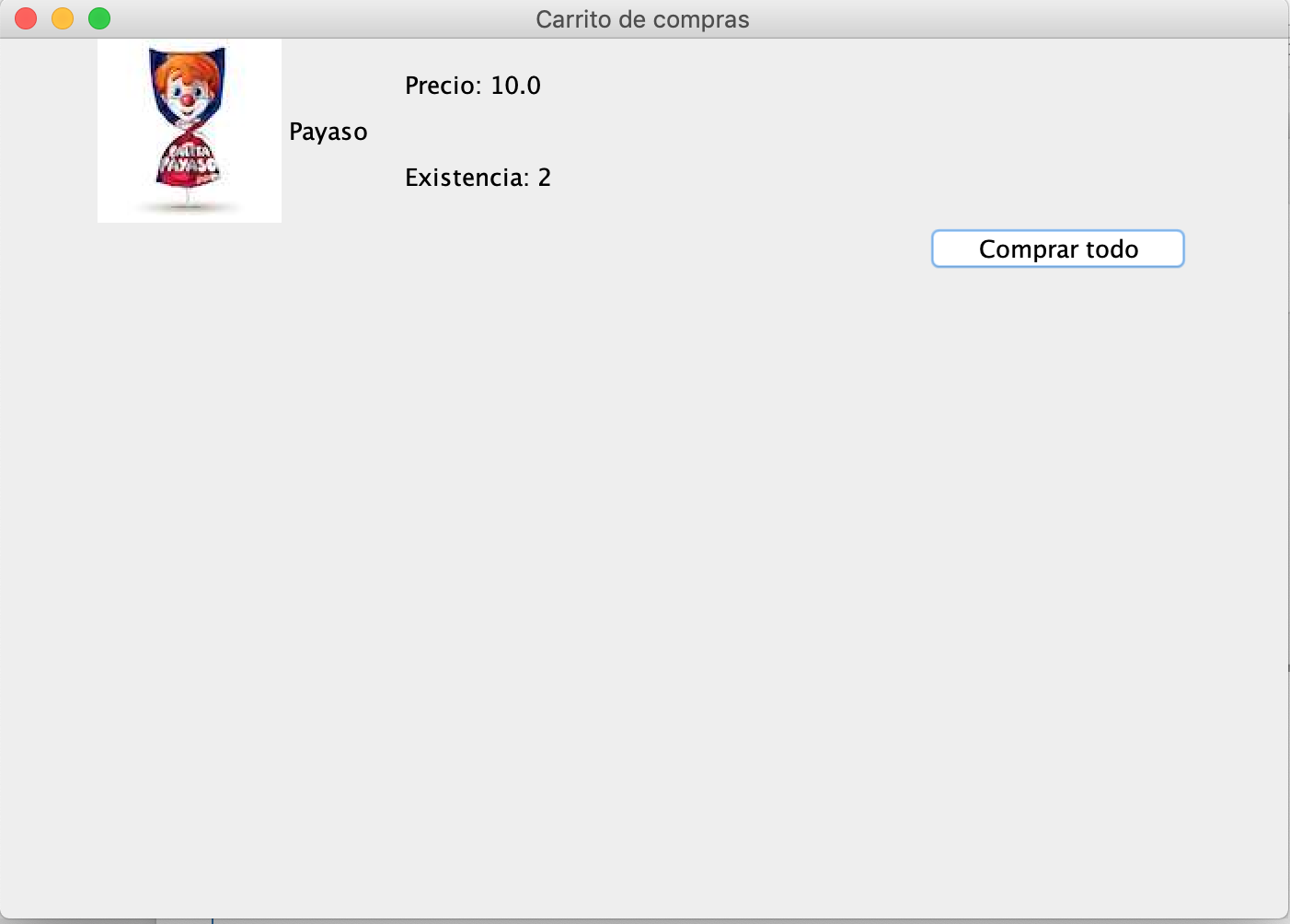
En está pantalla se pueden observar los distintos productos que están almacenados en la base de datos, así como su precio y existencia, cada uno de los productos tiene un botón llamado “Agregar al carrito” con el cual se puede agregar el producto seleccionado al carrito de compras.

Para acceder al carrito de compras y finalizar las compras se debe presionar el botón “Carrito”.



Carrito de compras:

En está pantalla se observan los productos que se hayan agregado al carrito de compras y al presionar el botón “Comprar todo” se finaliza la compra y la existencia de los productos disminuye.



Código servidor:

|  |
| --- |
| **import** **javax.swing.JFrame**;  **import** **java.net.ServerSocket**;  **import** **java.sql.ResultSet**;  **import** **java.io.ObjectInputStream**;  **import** **java.io.ObjectOutputStream**;  **import** **javax.swing.JLabel**;  **import** **javax.swing.JButton**;  **import** **javax.swing.ImageIcon**;  **import** **java.net.Socket**;  **public** **class** **Server**{  **public** **static** **final** int PUERTO=8000;  **public** **static** void main(String[] args) {  Producto[] compras;  JFrame frame=**new** JFrame("Servidor carrito de compras");  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  frame.setBounds(100, 100, 500, 500);  frame.setVisible(**true**);  **try**{  ResultSet rs;  cConexion conexion=**new** cConexion();  conexion.conectar();  rs=conexion.consulta("select count(\*) from producto;");  **if**(rs.first()){  JLabel[] etiquetas=**new** JLabel[rs.getInt(1)];  JLabel[] precios=**new** JLabel[rs.getInt(1)];  JLabel[] existencias=**new** JLabel[rs.getInt(1)];  Producto[] productos=**new** Producto[rs.getInt(1)];  rs=conexion.consulta("select \* from producto order by id;");  **if**(rs.first()){  **for**(int i=0; i<productos.length;i++){  productos[i]=**new** Producto(rs.getInt(1), rs.getString(2), rs.getFloat(3), rs.getInt(4), rs.getString(5));  etiquetas[i]=**new** JLabel(productos[i].getNombre(),**new** ImageIcon(productos[i].getImagen()),JLabel.RIGHT);  etiquetas[i].setBounds(0, 100\*i, 200, 100);  precios[i]=**new** JLabel("Precio: "+ productos[i].getPrecio());  precios[i].setBounds(220, 100\*i, 100, 50);  existencias[i]=**new** JLabel("Existencia: "+ productos[i].getExistencia());  existencias[i].setBounds(220, 50+(100\*i), 100, 50);  frame.add(etiquetas[i]);  frame.add(precios[i]);  frame.add(existencias[i]);  rs.next();  }  JLabel producto4=**new** JLabel("");  producto4.setBounds(0, 400, 80, 80);  frame.add(producto4);  *//Iniciamos el server aquí*  ServerSocket s=**new** ServerSocket(PUERTO);  System.out.println("Servidor iniciado, esperando clientes");  **while**(**true**){  Socket cl=s.accept();  System.out.println("Cliente conectado, empezando a transmitir información");  Integer cantidad=productos.length;  ObjectOutputStream oos = **new** ObjectOutputStream(cl.getOutputStream());  System.out.println("A continuación se enviará la cantidad de productos encontrados: "+productos.length);  oos.writeObject(cantidad);  oos.flush();  System.out.println("Cantidad enviada, ahora se empezarán a enviar los productos");  **for**(int i=0; i<productos.length;i++){  System.out.println("Se enviará el producto: "+productos[i].getNombre());  oos.writeObject(productos[i]);  oos.flush();  }  *//Aquí recibimos cualquier actualización*  ObjectInputStream ois=**new** ObjectInputStream(cl.getInputStream());  Integer total=(Integer)ois.readObject();  compras=**new** Producto[total];  System.out.println("Un cliente requiere realizar una compra");  **for**(int i=0; i<total;i++){  compras[i]=(Producto)ois.readObject();  }  System.out.println("Total de productos a comprar: "+compras.length);  *//Registro en base de datos*  **for**(int i=0;i<compras.length;i++){  conexion.actualizar("UPDATE producto SET existencia = existencia - 1 WHERE id = "+compras[i].getId()+";");  }  rs=conexion.consulta("select existencia from producto order by id;");  **if**(rs.first()){  **for**(int i=0;i<productos.length;i++){  existencias[i].setText("Existencia: "+rs.getInt(1));  productos[i].setExistencia(rs.getInt(1));  rs.next();  }  }  System.out.println("Existencias actualizadas");  cl.close();  oos.close();  ois.close();  }  }**else**{  System.out.println("No existen productos");  }  }**else**{  System.out.println("No existen productos");  }  } **catch** (Exception e) {  e.printStackTrace();  }  }  } |

Cliente:

|  |
| --- |
| **import** **java.awt.event.WindowEvent**;  **import** **javax.swing.JFrame**;  **import** **java.awt.event.ActionEvent**;  **import** **java.io.ObjectInputStream**;  **import** **java.io.ObjectOutputStream**;  **import** **javax.swing.JLabel**;  **import** **javax.swing.JButton**;  **import** **javax.swing.ImageIcon**;  **import** **java.net.Socket**;  **import** **javax.swing.JButton**;  **import** **java.awt.event.ActionListener**;  **import** **java.util.ArrayList**;  **import** **javax.swing.JButton**;  **import** **java.util.ArrayList**;  **import** **java.util.List**;  **public** **class** **Cliente** **implements** ActionListener{  **public** **static** **final** String HOST="localhost";  **public** **static** **final** int PUERTO=8000;  List<Producto> carrito=**new** ArrayList<>();  Producto[] productos;  JButton carritoCompras;  Socket cl;  JFrame frame;  **public** Cliente(){  frame=**new** JFrame("Cliente carrito de compras");  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  frame.setBounds(100, 100, 700, 500);  frame.setVisible(**true**);  **try** {  cl=**new** Socket(HOST,PUERTO);  System.out.println("Conexion exitosa con el servidor");  ObjectInputStream ois=**new** ObjectInputStream(cl.getInputStream());  Integer cantidad=(Integer)ois.readObject();  System.out.println("Cantidad de productos a recibir: "+cantidad);  JLabel[] etiquetas=**new** JLabel[cantidad];  JLabel[] precios=**new** JLabel[cantidad];  JLabel[] existencias=**new** JLabel[cantidad];  JButton[] botones=**new** JButton[cantidad];  productos=**new** Producto[cantidad];  carritoCompras=**new** JButton("Carrito: 0",**new** ImageIcon("images/carrito.png"));  carritoCompras.setBounds(600, 50, 150, 100);  carritoCompras.addActionListener(**this**);  carritoCompras.setName("carrito");  frame.add(carritoCompras);    **for**(int i=0; i<cantidad;i++){  productos[i]=(Producto)ois.readObject();  System.out.println("Se recibio el producto: "+productos[i].getNombre());  etiquetas[i]=**new** JLabel(productos[i].getNombre(),**new** ImageIcon(productos[i].getImagen()),JLabel.RIGHT);  etiquetas[i].setBounds(0, 100\*i, 200, 100);  precios[i]=**new** JLabel("Precio: "+ productos[i].getPrecio());  precios[i].setBounds(220, 100\*i, 100, 50);  existencias[i]=**new** JLabel("Existencia: "+ productos[i].getExistencia());  existencias[i].setBounds(220, 50+(100\*i), 150, 50);  botones[i]=**new** JButton("Agregar al carrito");  botones[i].setBounds(320, 35+(100\*i), 150, 30);  botones[i].setName(""+productos[i].getId());  botones[i].addActionListener(**this**);  frame.add(etiquetas[i]);  frame.add(precios[i]);  frame.add(existencias[i]);  frame.add(botones[i]);  }  JLabel producto4=**new** JLabel("");  producto4.setBounds(0, 400, 80, 80);  frame.add(producto4);  } **catch** (Exception e) {  e.printStackTrace();  }  }  **public** **static** void main(String[] args) {  **new** Cliente();  }  @Override  **public** void actionPerformed(ActionEvent e) {  String nombre=((JButton)(e.getSource())).getName();  **if** (nombre.equals("carrito")){  *//Creo nueva ventana de carrito*  **new** Carrito(carrito,cl);  *// carrito.clear();*  carritoCompras.setText("Carrito: "+carrito.size());  frame.setVisible(**false**);  }**else**{  int id=Integer.parseInt(nombre);  carrito.add(productos[id-1]);  carritoCompras.setText("Carrito: "+carrito.size());  *//System.out.println(carrito);*  }  }  } |

Carrito de compras:

|  |
| --- |
| **import** **javax.swing.JFrame**;  **import** **java.awt.event.ActionEvent**;  **import** **java.io.ObjectInputStream**;  **import** **java.io.ObjectOutputStream**;  **import** **javax.swing.JLabel**;  **import** **javax.swing.JButton**;  **import** **javax.swing.ImageIcon**;  **import** **java.net.Socket**;  **import** **java.awt.event.ActionListener**;  **import** **java.util.ArrayList**;  **import** **java.util.List**;  **import** **java.util.stream.Stream**;  **import** **javax.swing.JOptionPane**;  **import** **java.lang.Thread**;  **public** **class** **Carrito** **implements** ActionListener{  Socket cl;  List<Producto> carrito;  JFrame frame;  **public** Carrito(List<Producto> carrito, Socket cl){  **this**.cl=cl;  **this**.carrito=carrito;  frame=**new** JFrame("Carrito de compras");  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  frame.setBounds(100, 100, 700, 500);  frame.setVisible(**true**);  JLabel[] etiquetas=**new** JLabel[carrito.size()];  JLabel[] precios=**new** JLabel[carrito.size()];  JLabel[] existencias=**new** JLabel[carrito.size()];  JButton comprar=**new** JButton("Comprar todo");  comprar.setBounds(500, 100, 150, 30);  comprar.addActionListener(**this**);  frame.add(comprar);  **for**(int i=0; i<carrito.size();i++){  etiquetas[i]=**new** JLabel(carrito.get(i).getNombre(),**new** ImageIcon(carrito.get(i).getImagen()),JLabel.RIGHT);  etiquetas[i].setBounds(0, 100\*i, 200, 100);  precios[i]=**new** JLabel("Precio: "+ carrito.get(i).getPrecio());  precios[i].setBounds(220, 100\*i, 100, 50);  existencias[i]=**new** JLabel("Existencia: "+ carrito.get(i).getExistencia());  existencias[i].setBounds(220, 50+(100\*i), 200, 50);  frame.add(etiquetas[i]);  frame.add(precios[i]);  frame.add(existencias[i]);  }  JLabel producto4=**new** JLabel("");  producto4.setBounds(0, 400, 80, 80);  frame.add(producto4);  }  @Override  **public** void actionPerformed(ActionEvent e) {  **try**{  System.out.println("Se ha iniciado el proceso de compra");  ObjectOutputStream oos = **new** ObjectOutputStream(cl.getOutputStream());  Integer total=carrito.size();  oos.writeObject(total);  oos.flush();  **for**(int i=0;i<carrito.size();i++){  oos.writeObject(carrito.get(i));  oos.flush();  }  JOptionPane.showMessageDialog(**null**, "Compra exitosa !");  frame.setVisible(**false**);    **try** {  cl.close();  oos.close();  } **catch** (Exception ex) {  ex.printStackTrace();  }  System.exit(0);  *//new Cliente();*    }**catch**(Exception ex) {  ex.printStackTrace();  }  }  } |

# Conclusiones

El trabajar con sockets abre nuevas posibilidades a nosotros como desarrolladores para crear aplicaciones que puedan comunicarse entre si, en está práctica usamos el servidor y los clientes en la misma computadora, pero en caso de requerirlo, se pudo tener el servidor en una computadora y el cliente en otra, y solo hubiera sido necesario cambiar la ip de la conexión por parte del cliente.