|  |  |  |
| --- | --- | --- |
|  | **INSTITUTO POLITECNICO NACIONAL**  Escuela Superior de Cómputo |  |

Alumnos:

Guerra Vargas Irving Cristobal

Pantoja Flores Pablo

Unidad de Aprendizaje: Aplicaciones para Comunicaciones en Red

Profesor: Axel Ernesto Moreno Cervantes

**PRACTICA 2: Tienda**

Grupo: 3CM7

Contenido

[Introducción 3](#_Toc516597762)

[Desarrollo 3](#_Toc516597763)

[Códigos 5](#_Toc516597764)

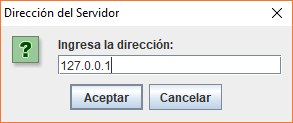
[Conclusiones 31](#_Toc516597765)

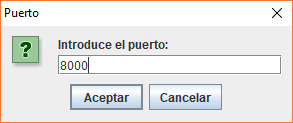
# Introducción

En esta práctica se simula el funcionamiento de una tienda online en donde el catálogo de los productos en venta se encuentra del lado del servidor. El objetivo de esta práctica es aprender a manejar el envío de objetos. Desde el envío de imágenes hasta los datos que contiene cada producto (la clase que represente cada producto).

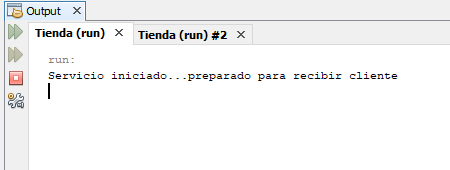
# Desarrollo

La aplicación comienza solicitándole al cliente el puerto y la dirección en la cual se encuentra el servidor.

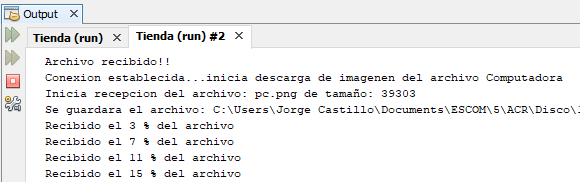


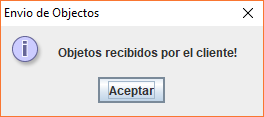


Posteriormente, iniciamos el servidor que está a la espera de clientes para enviarle los objetos correspondientes.



El proceso se repite para cada uno de los archivos y el usuario es notificado del envío de todos los productos del catálogo al cliente.

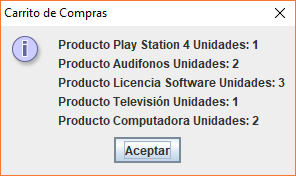




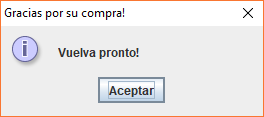
Después se muestra al usuario la interfaz en la cual podrá agregar productos a su carrito de compras y finalizar su compra.

La actualización de las existencias en el servidor se realiza siempre que el usuario termine una compra.





Termina la aplicación.



# Códigos

**Carrito.java**

**/\***

**\* To change this license header, choose License Headers in Project Properties.**

**\* To change this template file, choose Tools | Templates**

**\* and open the template in the editor.**

**\*/**

**package escomarket;**

**import java.io.Serializable;**

**/\*\***

**\***

**\* @author irvingmac**

**\*/**

**public class Carrito implements Serializable{**

**private int Producto;**

**private int cantidad;**

**public Carrito(int Producto, int cantidad) {**

**this.Producto = Producto;**

**this.cantidad = cantidad;**

**}**

**public Carrito(){**

**this.Producto = 0;**

**this.cantidad = 0;**

**}**

**public int getProducto() {**

**return Producto;**

**}**

**public void setProducto(int Producto) {**

**this.Producto = Producto;**

**}**

**public int getCantidad() {**

**return cantidad;**

**}**

**public void setCantidad(int cantidad) {**

**this.cantidad = cantidad;**

**}**

**}**

**Cliente.java**

**/\***

**\* To change this license header, choose License Headers in Project Properties.**

**\* To change this template file, choose Tools | Templates**

**\* and open the template in the editor.**

**\*/**

**package escomarket;**

**import java.awt.Image;**

**import java.io.DataInputStream;**

**import java.io.DataOutputStream;**

**import java.io.File;**

**import java.io.FileOutputStream;**

**import java.io.ObjectInputStream;**

**import java.io.ObjectOutputStream;**

**import java.io.OutputStreamWriter;**

**import java.io.PrintWriter;**

**import java.net.Socket;**

**import java.util.ArrayList;**

**import javax.swing.DefaultListModel;**

**import javax.swing.Icon;**

**import javax.swing.ImageIcon;**

**import static javax.swing.JOptionPane.showMessageDialog;**

**/\*\***

**\***

**\* @author irvingmac**

**\*/**

**public class Cliente extends javax.swing.JFrame {**

**int puerto = 1234;**

**String host="127.0.0.1";**

**int x,y;**

**//Socket**

**Socket cl;**

**ArrayList<Producto> Inventario = new ArrayList<>(5);**

**//String pathImg = "/Volumes/BLUE/ESCOM/7/Redes2/Practica 2 - Tienda/Escomarket/src/imgCliente";**

**//String pathServ = "/Volumes/BLUE/ESCOM/7/Redes2/Practica 2 - Tienda/Escomarket/src/imgServidor";**

**String pathImg = "F:\\ESCOM\\7\\Redes2\\Practica 2 - Tienda\\Escomarket\\src\\imgCliente";**

**String pathServ = "F:\\ESCOM\\7\\Redes2\\Practica 2 - Tienda\\Escomarket\\src\\imgServidor";**

**Producto productoSelected = null;**

**ArrayList<Carrito> myCarrito = new ArrayList<>(5);**

**/\*\***

**\* Creates new form Cliente**

**\*/**

**public Cliente() {**

**initComponents();**

**this.setLocationRelativeTo(null);**

**//Cargamos el inventario o catalogo**

**Inventario = CargarInventario();**

**loadImagenes();**

**loadInfoInventarioJFRAME();**

**}**

**/\*\***

**\* This method is called from within the constructor to initialize the form.**

**\* WARNING: Do NOT modify this code. The content of this method is always**

**\* regenerated by the Form Editor.**

**\*/**

**@SuppressWarnings("unchecked")**

**// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents**

**private void initComponents() {**

**btnSalir = new javax.swing.JButton();**

**jLabel1 = new javax.swing.JLabel();**

**jScrollPane1 = new javax.swing.JScrollPane();**

**InventarioList = new javax.swing.JList<>();**

**jLabel2 = new javax.swing.JLabel();**

**jLabel3 = new javax.swing.JLabel();**

**jLabel4 = new javax.swing.JLabel();**

**jLabel5 = new javax.swing.JLabel();**

**jLabel6 = new javax.swing.JLabel();**

**jLabel7 = new javax.swing.JLabel();**

**nombreSelected = new javax.swing.JLabel();**

**marcaSelected = new javax.swing.JLabel();**

**descripcionSelected = new javax.swing.JLabel();**

**precioSelected = new javax.swing.JLabel();**

**existSelected = new javax.swing.JLabel();**

**lblIMG = new javax.swing.JLabel();**

**btnAddCarrito = new javax.swing.JButton();**

**btnDeleteCarrito = new javax.swing.JButton();**

**btnVaciar = new javax.swing.JButton();**

**btnVer = new javax.swing.JButton();**

**btnFin = new javax.swing.JButton();**

**FondoCliente = new javax.swing.JLabel();**

**setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);**

**setUndecorated(true);**

**getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());**

**btnSalir.setText("Salir");**

**btnSalir.addActionListener(new java.awt.event.ActionListener() {**

**public void actionPerformed(java.awt.event.ActionEvent evt) {**

**btnSalirActionPerformed(evt);**

**}**

**});**

**getContentPane().add(btnSalir, new org.netbeans.lib.awtextra.AbsoluteConstraints(630, 10, -1, -1));**

**jLabel1.setText("Productos disponibles");**

**getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 10, 150, 30));**

**InventarioList.setModel(new javax.swing.AbstractListModel<String>() {**

**String[] strings = { "Item 1", "Item 2", "Item 3", "Item 4", "Item 5" };**

**public int getSize() { return strings.length; }**

**public String getElementAt(int i) { return strings[i]; }**

**});**

**InventarioList.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mouseClicked(java.awt.event.MouseEvent evt) {**

**InventarioListMouseClicked(evt);**

**}**

**});**

**jScrollPane1.setViewportView(InventarioList);**

**getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 50, 670, 90));**

**jLabel2.setText("Producto seleccionado");**

**getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 160, -1, 20));**

**jLabel3.setText("Nombre: ");**

**getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 190, -1, -1));**

**jLabel4.setText("Marca:");**

**getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 220, -1, -1));**

**jLabel5.setText("Descripcion:");**

**getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 250, -1, -1));**

**jLabel6.setText("Precio:");**

**getContentPane().add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 280, -1, -1));**

**jLabel7.setText("Existencias:");**

**getContentPane().add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 310, -1, 20));**

**nombreSelected.setText("Sin seleccionar");**

**getContentPane().add(nombreSelected, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 190, 200, 20));**

**marcaSelected.setText("Sin seleccionar");**

**getContentPane().add(marcaSelected, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 220, -1, -1));**

**descripcionSelected.setText("Sin seleccionar");**

**getContentPane().add(descripcionSelected, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 250, -1, -1));**

**precioSelected.setText("Sin seleccionar");**

**getContentPane().add(precioSelected, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 280, -1, -1));**

**existSelected.setText("Sin seleccionar");**

**getContentPane().add(existSelected, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 310, -1, -1));**

**lblIMG.setText("IMAGEN");**

**getContentPane().add(lblIMG, new org.netbeans.lib.awtextra.AbsoluteConstraints(360, 150, 200, 180));**

**btnAddCarrito.setText("+");**

**btnAddCarrito.addActionListener(new java.awt.event.ActionListener() {**

**public void actionPerformed(java.awt.event.ActionEvent evt) {**

**btnAddCarritoActionPerformed(evt);**

**}**

**});**

**getContentPane().add(btnAddCarrito, new org.netbeans.lib.awtextra.AbsoluteConstraints(630, 150, 40, 30));**

**btnDeleteCarrito.setText("-");**

**btnDeleteCarrito.addActionListener(new java.awt.event.ActionListener() {**

**public void actionPerformed(java.awt.event.ActionEvent evt) {**

**btnDeleteCarritoActionPerformed(evt);**

**}**

**});**

**getContentPane().add(btnDeleteCarrito, new org.netbeans.lib.awtextra.AbsoluteConstraints(630, 180, 40, 30));**

**btnVaciar.setText("Vaciar Carrito");**

**btnVaciar.addActionListener(new java.awt.event.ActionListener() {**

**public void actionPerformed(java.awt.event.ActionEvent evt) {**

**btnVaciarActionPerformed(evt);**

**}**

**});**

**getContentPane().add(btnVaciar, new org.netbeans.lib.awtextra.AbsoluteConstraints(570, 220, 130, 30));**

**btnVer.setText("Ver Carrito");**

**btnVer.addActionListener(new java.awt.event.ActionListener() {**

**public void actionPerformed(java.awt.event.ActionEvent evt) {**

**btnVerActionPerformed(evt);**

**}**

**});**

**getContentPane().add(btnVer, new org.netbeans.lib.awtextra.AbsoluteConstraints(570, 250, 130, 30));**

**btnFin.setText("Finalizar Compra");**

**btnFin.addActionListener(new java.awt.event.ActionListener() {**

**public void actionPerformed(java.awt.event.ActionEvent evt) {**

**btnFinActionPerformed(evt);**

**}**

**});**

**getContentPane().add(btnFin, new org.netbeans.lib.awtextra.AbsoluteConstraints(570, 280, 130, 30));**

**FondoCliente.setIcon(new javax.swing.ImageIcon(getClass().getResource("/img/wall1.jpg"))); // NOI18N**

**FondoCliente.setText("jLabel1");**

**FondoCliente.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {**

**public void mouseDragged(java.awt.event.MouseEvent evt) {**

**FondoClienteMouseDragged(evt);**

**}**

**});**

**FondoCliente.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mousePressed(java.awt.event.MouseEvent evt) {**

**FondoClienteMousePressed(evt);**

**}**

**});**

**getContentPane().add(FondoCliente, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 710, 370));**

**pack();**

**}// </editor-fold>//GEN-END:initComponents**

**private void btnSalirActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnSalirActionPerformed**

**System.exit(0);**

**}//GEN-LAST:event\_btnSalirActionPerformed**

**private void FondoClienteMousePressed(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_FondoClienteMousePressed**

**x = evt.getX();**

**y = evt.getY();**

**}//GEN-LAST:event\_FondoClienteMousePressed**

**private void FondoClienteMouseDragged(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_FondoClienteMouseDragged**

**this.setLocation(this.getLocation().x + evt.getX() -x,this.getLocation().y + evt.getY() -y);**

**}//GEN-LAST:event\_FondoClienteMouseDragged**

**private void InventarioListMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_InventarioListMouseClicked**

**Object sel =null;**

**int[] selectedIx = this.InventarioList.getSelectedIndices();**

**for (int i = 0; i < selectedIx.length; i++) {**

**sel = InventarioList.getModel().getElementAt(selectedIx[i]);**

**}**

**String aux = sel.toString();**

**String[] ind = aux.split("-");**

**int indice = Integer.valueOf(ind[0]);**

**productoSelected = Inventario.get(indice);**

**nombreSelected.setText(productoSelected.getNombre());**

**marcaSelected.setText(productoSelected.getMarca());**

**precioSelected.setText(""+productoSelected.getPrecio());**

**descripcionSelected.setText(productoSelected.getDescripcion());**

**existSelected.setText(""+productoSelected.getExistencias());**

**String nameFile = "/imgCliente"+productoSelected.getImagen().toString().replace(pathServ, "");**

**ImageIcon imagen = new javax.swing.ImageIcon(getClass().getResource(nameFile));**

**Icon icono = new javax.swing.ImageIcon(imagen.getImage().getScaledInstance(lblIMG.getWidth(),lblIMG.getHeight(),Image.SCALE\_DEFAULT));**

**lblIMG.setIcon(icono);**

**}//GEN-LAST:event\_InventarioListMouseClicked**

**private void btnAddCarritoActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnAddCarritoActionPerformed**

**if(productoSelected != null){**

**int indiceProducto = productoSelected.getID();**

**int existencias = Inventario.get(indiceProducto).getExistencias();**

**if(existencias>0){**

**Inventario.get(indiceProducto).setExistencias(existencias-1);**

**existSelected.setText(""+Inventario.get(indiceProducto).getExistencias());**

**if (existeEnCarrito(myCarrito,indiceProducto) == false){**

**System.out.println("No existe");**

**Carrito aux = new Carrito(indiceProducto,1);**

**myCarrito.add(aux);**

**}else{**

**System.out.println("Existe");**

**int indice = getIndiceExisteEnCarrito(myCarrito,indiceProducto);**

**int exits = myCarrito.get(indice).getCantidad();**

**myCarrito.get(indice).setCantidad(exits+1);**

**System.out.println(myCarrito.get(indice).getCantidad());**

**}**

**}else{**

**showMessageDialog(null, "[ FAIL ] No hay existencias del producto");**

**}**

**}else{**

**showMessageDialog(null, "[ FAIL ] Seleccione un producto para agregar o eliminar del carrito");**

**}**

**}//GEN-LAST:event\_btnAddCarritoActionPerformed**

**private void btnDeleteCarritoActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnDeleteCarritoActionPerformed**

**if(productoSelected != null){**

**int indiceProducto = productoSelected.getID();**

**int existencias = Inventario.get(indiceProducto).getExistencias();**

**if (existeEnCarrito(myCarrito,indiceProducto) == false){**

**//nada, no tiene ni mais**

**}else{**

**System.out.println("Existe");**

**int indice = getIndiceExisteEnCarrito(myCarrito,indiceProducto);**

**int exits = myCarrito.get(indice).getCantidad();**

**if(exits>0){**

**myCarrito.get(indice).setCantidad(exits-1);**

**System.out.println(myCarrito.get(indice).getCantidad());**

**Inventario.get(indiceProducto).setExistencias(existencias+1);**

**existSelected.setText(""+Inventario.get(indiceProducto).getExistencias());**

**}**

**}**

**}else{**

**showMessageDialog(null, "[ FAIL ] Seleccione un producto para agregar o eliminar del carrito");**

**}**

**}//GEN-LAST:event\_btnDeleteCarritoActionPerformed**

**private void btnVaciarActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnVaciarActionPerformed**

**Inventario = CargarInventario();**

**loadInfoInventarioJFRAME();**

**productoSelected = Inventario.get(productoSelected.getID());**

**nombreSelected.setText(productoSelected.getNombre());**

**marcaSelected.setText(productoSelected.getMarca());**

**precioSelected.setText(""+productoSelected.getPrecio());**

**descripcionSelected.setText(productoSelected.getDescripcion());**

**existSelected.setText(""+productoSelected.getExistencias());**

**vaciarCarrito();**

**}//GEN-LAST:event\_btnVaciarActionPerformed**

**private void btnVerActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnVerActionPerformed**

**String misCosas = "";**

**for(int i = 0;i<myCarrito.size();i++){**

**misCosas += Inventario.get(myCarrito.get(i).getProducto()).getNombre() + " Cantidad: "+myCarrito.get(i).getCantidad()+"\n";**

**}**

**showMessageDialog(null, "[ OK ] Mostrando inventario...\n\n"+misCosas);**

**}//GEN-LAST:event\_btnVerActionPerformed**

**private void btnFinActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnFinActionPerformed**

**if(myCarrito.isEmpty()){**

**showMessageDialog(null, "[ FAIL ] Tu carrito esta vacio");**

**}else{**

**connectWithServer();**

**try{**

**PrintWriter pw = new PrintWriter(new OutputStreamWriter(cl.getOutputStream()));**

**pw.println("3"); //Le mandamos un 3, indicando que queremos comprar**

**pw.flush();**

**System.out.println("[ OK ] Preparado para comprar...");**

**Producto p;**

**ObjectOutputStream obj = new ObjectOutputStream(cl.getOutputStream());**

**obj.writeObject(myCarrito);**

**obj.flush();**

**System.out.println("[ OK ] Carrito de comnpras enviado!");**

**obj.close();**

**pw.close();**

**cl.close();**

**}catch(Exception e){**

**showMessageDialog(null, "[ FAIL ] Ocurrio un problema al recibir los productos");**

**}**

**showMessageDialog(null, "[ OK ] Gracias por comprar con nosotros");**

**Inventario = CargarInventario();**

**loadInfoInventarioJFRAME();**

**productoSelected = Inventario.get(productoSelected.getID());**

**nombreSelected.setText(productoSelected.getNombre());**

**marcaSelected.setText(productoSelected.getMarca());**

**precioSelected.setText(""+productoSelected.getPrecio());**

**descripcionSelected.setText(productoSelected.getDescripcion());**

**existSelected.setText(""+productoSelected.getExistencias());**

**vaciarCarrito();**

**}**

**}//GEN-LAST:event\_btnFinActionPerformed**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String args[]) {**

**/\* Set the Nimbus look and feel \*/**

**//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">**

**/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.**

**\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html**

**\*/**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

**java.util.logging.Logger.getLogger(Cliente.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (InstantiationException ex) {**

**java.util.logging.Logger.getLogger(Cliente.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (IllegalAccessException ex) {**

**java.util.logging.Logger.getLogger(Cliente.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

**java.util.logging.Logger.getLogger(Cliente.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**}**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**new Cliente().setVisible(true);**

**}**

**});**

**}**

**private void connectWithServer() {**

**//Nos conectamos con els ervidor local.**

**try{**

**cl = new Socket (host, puerto); //socket bloquante**

**System.out.println("[ OK ] Conectado!");**

**}catch(Exception e){**

**showMessageDialog(null, "[ FAIL ] Ocurrio un problema con el servidor");**

**}**

**}**

**// Variables declaration - do not modify//GEN-BEGIN:variables**

**private javax.swing.JLabel FondoCliente;**

**private javax.swing.JList<String> InventarioList;**

**private javax.swing.JButton btnAddCarrito;**

**private javax.swing.JButton btnDeleteCarrito;**

**private javax.swing.JButton btnFin;**

**private javax.swing.JButton btnSalir;**

**private javax.swing.JButton btnVaciar;**

**private javax.swing.JButton btnVer;**

**private javax.swing.JLabel descripcionSelected;**

**private javax.swing.JLabel existSelected;**

**private javax.swing.JLabel jLabel1;**

**private javax.swing.JLabel jLabel2;**

**private javax.swing.JLabel jLabel3;**

**private javax.swing.JLabel jLabel4;**

**private javax.swing.JLabel jLabel5;**

**private javax.swing.JLabel jLabel6;**

**private javax.swing.JLabel jLabel7;**

**private javax.swing.JScrollPane jScrollPane1;**

**private javax.swing.JLabel lblIMG;**

**private javax.swing.JLabel marcaSelected;**

**private javax.swing.JLabel nombreSelected;**

**private javax.swing.JLabel precioSelected;**

**// End of variables declaration//GEN-END:variables**

**private ArrayList<Producto> CargarInventario() {**

**ArrayList<Producto> Catalogo = new ArrayList<>(5);**

**connectWithServer();**

**try{**

**PrintWriter pw = new PrintWriter(new OutputStreamWriter(cl.getOutputStream()));**

**pw.println("1"); //Le mandamos un 1, indicando que queremos cargar objetos**

**pw.flush();**

**System.out.println("[ OK ] Preparado para recibir objetos...");**

**Producto p;**

**ObjectInputStream ois = new ObjectInputStream(cl.getInputStream());**

**for(int i=0;i<5;i++){**

**p = (Producto) ois.readObject();**

**Catalogo.add(i, p);**

**System.out.println("[ OK ] Objeto recibido: " +**

**Catalogo.get(i).getID() + "," +**

**Catalogo.get(i).getNombre() + "," +**

**Catalogo.get(i).getMarca() + "," +**

**Catalogo.get(i).getDescripcion() + "," +**

**Catalogo.get(i).getPrecio());**

**}**

**ois.close();**

**cl.close();**

**}catch(Exception e){**

**showMessageDialog(null, "[ FAIL ] Ocurrio un problema al recibir los productos");**

**}**

**System.out.println("[ OK ] Cantidad de productos recibidos: "+Catalogo.size());**

**return Catalogo;**

**}**

**private void loadImagenes() {**

**System.out.println("[ OK ] Cargando imagenes de productos");**

**for(int j=0;j<Inventario.size();j++){**

**DescargaImagen(Inventario.get(j).getNombre());**

**}**

**}**

**private void DescargaImagen(String nombreImg) {**

**connectWithServer();**

**try{**

**PrintWriter pw = new PrintWriter(new OutputStreamWriter(cl.getOutputStream()));**

**byte[] b = new byte[1500];**

**long recibidos = 0;**

**int n,porcentaje = 0;**

**pw.println("2");**

**pw.flush();**

**System.out.println("[ OK ] Inicia descarga de imagenen del prodcuto "+nombreImg);**

**pw.println(nombreImg);**

**pw.flush();**

**DataInputStream dis = new DataInputStream(cl.getInputStream());**

**String nombres = dis.readUTF();**

**long tam = dis.readLong();**

**System.out.println("[ OK ] Iniciando recepcion del archivo: "+nombres+" de tamaño: "+tam);**

**File ff= new File(pathImg);**

**String pathDest = ff.getAbsolutePath();**

**DataOutputStream dos = new DataOutputStream(new FileOutputStream(pathDest+"/"+nombres));**

**System.out.println("[ OK ] Se guardara el archivo: "+pathDest+"/"+nombres);**

**while(recibidos<tam){**

**n = dis.read(b);**

**recibidos+=n;**

**dos.write(b,0,n);**

**dos.flush();**

**porcentaje = (int)((recibidos\*100)/tam);**

**System.out.println("[ OK ] Recibido el "+porcentaje+" %");**

**}**

**System.out.println("[ OK ] Imagen recibida con exito.");**

**dos.close();**

**dis.close();**

**pw.close();**

**cl.close();**

**}catch(Exception e){**

**showMessageDialog(null, "[ FAIL ] Ocurrio un problema al recibir las imagenes");**

**}**

**}**

**private void loadInfoInventarioJFRAME() {**

**String[] data = new String [Inventario.size()];**

**for(int i = 0;i<Inventario.size();i++){**

**data[i] = i+"-"+Inventario.get(i).getNombre();**

**}**

**InventarioList.setListData(data);**

**}**

**private boolean existeEnCarrito(ArrayList<Carrito> myCarrito, int indiceProducto) {**

**for(Carrito p:myCarrito)if (p.getProducto() == indiceProducto)return true;return false;**

**}**

**private int getIndiceExisteEnCarrito(ArrayList<Carrito> myCarrito, int indiceProducto) {**

**for(int i = 0;i<myCarrito.size();i++){**

**if (myCarrito.get(i).getProducto()==indiceProducto){**

**return i;**

**}**

**}**

**return 0;**

**}**

**private void vaciarCarrito() {**

**while(myCarrito.isEmpty()==false){**

**myCarrito.remove(0);**

**}**

**}**

**}**

**Producto.java**

**/\***

**\* To change this license header, choose License Headers in Project Properties.**

**\* To change this template file, choose Tools | Templates**

**\* and open the template in the editor.**

**\*/**

**package escomarket;**

**import java.io.File;**

**import java.io.Serializable;**

**/\*\***

**\***

**\* @author irvingmac**

**\*/**

**class Producto implements Serializable{**

**int ID;**

**String Nombre;**

**String Marca;**

**String Descripcion;**

**int Precio;**

**int Existencias;**

**File Imagen;**

**//Ubicación de las imagenes de los productos**

**//Rutas WIN**

**//private final String pathImg = "F:\\ESCOM\\7\\Redes2\\Practica 1\\FilesServer\\src\\cloud";**

**//Rutas MAC**

**//private final String pathImg = "/Volumes/BLUE/ESCOM/7/Redes2/Practica 2 - Tienda/Escomarket/src/imgServidor";**

**private final String pathImg = "F:\\ESCOM\\7\\Redes2\\Practica 2 - Tienda\\Escomarket\\src\\imgServidor";**

**public Producto(int ID, String Nombre, String Marca, String Descripcion, int Precio, int Existencias, String Imagen) {**

**this.ID = ID;**

**this.Nombre = Nombre;**

**this.Marca = Marca;**

**this.Descripcion = Descripcion;**

**this.Precio = Precio;**

**this.Existencias = Existencias;**

**this.Imagen = new File(pathImg, Imagen);**

**}**

**public Producto() {**

**}**

**public int getID() {**

**return ID;**

**}**

**public void setID(int ID) {**

**this.ID = ID;**

**}**

**public String getNombre() {**

**return Nombre;**

**}**

**public void setNombre(String Nombre) {**

**this.Nombre = Nombre;**

**}**

**public String getMarca() {**

**return Marca;**

**}**

**public void setMarca(String Marca) {**

**this.Marca = Marca;**

**}**

**public String getDescripcion() {**

**return Descripcion;**

**}**

**public void setDescripcion(String Descripcion) {**

**this.Descripcion = Descripcion;**

**}**

**public int getPrecio() {**

**return Precio;**

**}**

**public void setPrecio(int Precio) {**

**this.Precio = Precio;**

**}**

**public int getExistencias() {**

**return Existencias;**

**}**

**public void setExistencias(int Existencias) {**

**this.Existencias = Existencias;**

**}**

**public File getImagen() {**

**return Imagen;**

**}**

**public void setImagen(File Imagen) {**

**this.Imagen = Imagen;**

**}**

**}**

**Servidor.java**

**/\***

**\* To change this license header, choose License Headers in Project Properties.**

**\* To change this template file, choose Tools | Templates**

**\* and open the template in the editor.**

**\*/**

**package escomarket;**

**import java.io.BufferedReader;**

**import java.io.DataInputStream;**

**import java.io.DataOutputStream;**

**import java.io.File;**

**import java.io.FileInputStream;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**import java.io.ObjectInputStream;**

**import java.io.ObjectOutputStream;**

**import java.net.ServerSocket;**

**import java.net.Socket;**

**import java.util.ArrayList;**

**/\*\***

**\***

**\* @author irvingmac**

**\*/**

**public class Servidor extends javax.swing.JFrame {**

**int puerto = 1234;**

**String host = "127.0.0.1";**

**ServerSocket s;**

**Socket cl;**

**int x,y;**

**ArrayList<Producto> Inventario = new ArrayList(5);**

**//Respaldo de existencias iniciales.**

**private static int AUX[] = new int[5];**

**Carrito clienteCarrito = new Carrito();**

**ArrayList<Carrito> clienteCarritoLista = new ArrayList<>(5);**

**/\*\***

**\* Creates new form Servidor**

**\*/**

**public Servidor() {**

**initComponents();**

**this.setLocationRelativeTo(null);**

**startService();**

**cargarProductos();**

**esperaCliente();**

**}**

**/\*\***

**\* This method is called from within the constructor to initialize the form.**

**\* WARNING: Do NOT modify this code. The content of this method is always**

**\* regenerated by the Form Editor.**

**\*/**

**@SuppressWarnings("unchecked")**

**// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents**

**private void initComponents() {**

**jLabel2 = new javax.swing.JLabel();**

**btnSalir = new javax.swing.JButton();**

**FondoServidor = new javax.swing.JLabel();**

**setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);**

**setUndecorated(true);**

**getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());**

**jLabel2.setText("ADMINISTRACION");**

**getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 20, 120, 20));**

**btnSalir.setText("Salir");**

**btnSalir.addActionListener(new java.awt.event.ActionListener() {**

**public void actionPerformed(java.awt.event.ActionEvent evt) {**

**btnSalirActionPerformed(evt);**

**}**

**});**

**getContentPane().add(btnSalir, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 20, -1, -1));**

**FondoServidor.setIcon(new javax.swing.ImageIcon(getClass().getResource("/img/wall1.jpg"))); // NOI18N**

**FondoServidor.setText("jLabel1");**

**FondoServidor.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {**

**public void mouseDragged(java.awt.event.MouseEvent evt) {**

**FondoServidorMouseDragged(evt);**

**}**

**});**

**FondoServidor.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mousePressed(java.awt.event.MouseEvent evt) {**

**FondoServidorMousePressed(evt);**

**}**

**});**

**getContentPane().add(FondoServidor, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 630, 370));**

**pack();**

**}// </editor-fold>//GEN-END:initComponents**

**private void btnSalirActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnSalirActionPerformed**

**System.exit(0);**

**}//GEN-LAST:event\_btnSalirActionPerformed**

**private void FondoServidorMousePressed(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_FondoServidorMousePressed**

**x = evt.getX();**

**y = evt.getY();**

**}//GEN-LAST:event\_FondoServidorMousePressed**

**private void FondoServidorMouseDragged(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_FondoServidorMouseDragged**

**this.setLocation(this.getLocation().x + evt.getX() -x,this.getLocation().y + evt.getY() -y);**

**}//GEN-LAST:event\_FondoServidorMouseDragged**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String args[]) {**

**/\* Set the Nimbus look and feel \*/**

**//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">**

**/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.**

**\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html**

**\*/**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

**java.util.logging.Logger.getLogger(Servidor.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (InstantiationException ex) {**

**java.util.logging.Logger.getLogger(Servidor.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (IllegalAccessException ex) {**

**java.util.logging.Logger.getLogger(Servidor.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

**java.util.logging.Logger.getLogger(Servidor.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**}**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**new Servidor().setVisible(true);**

**}**

**});**

**}**

**private void startService() {**

**try{**

**s = new ServerSocket(puerto);**

**s.setReuseAddress(true);**

**System.out.println("[ OK ] Servicio Iniciado, esperando cliente...");**

**} catch(Exception e){**

**e.printStackTrace();**

**}**

**}**

**// Variables declaration - do not modify//GEN-BEGIN:variables**

**private javax.swing.JLabel FondoServidor;**

**private javax.swing.JButton btnSalir;**

**private javax.swing.JLabel jLabel2;**

**// End of variables declaration//GEN-END:variables**

**private void cargarProductos() {**

**Producto p1 = new Producto(0, "Play Station 4", "Sony", "Consola de Videojuegos", 7599, 20, "ps4.jpg");**

**Producto p2 = new Producto(1, "Audifonos", "Hissense", "Audifonos inalámbricos", 1100, 18, "audifonos.jpg");**

**Producto p3 = new Producto(2, "Licencia Software", "Microsoft", "Licencia Windows 10 Home Edition 1 año", 1200, 30, "software.jpg");**

**Producto p4 = new Producto(3, "Televisión", "Samsung", "Smart TV Samsung 32 pulg. HD", 4500, 15, "tv.jpg");**

**Producto p5 = new Producto(4, "Computadora", "DELL", "Computadora de escritorio DELL Inspiron 45700", 8699, 25, "pc.jpg");**

**System.out.println("[ OK ] Productos creados");**

**Inventario.add(0, p1);**

**Inventario.add(1, p2);**

**Inventario.add(2, p3);**

**Inventario.add(3, p4);**

**Inventario.add(4, p5);**

**System.out.println("[ OK ] Productos añadidos correctamente al inventario");**

**for (int t = 0; t < Inventario.size(); t++) {**

**AUX[t] = Inventario.get(t).getExistencias(); //Si desea limpiar el carrito**

**}**

**}**

**private void esperaCliente() {**

**try{**

**while (true) {**

**Socket cl = s.accept();**

**BufferedReader br = new BufferedReader(new InputStreamReader(cl.getInputStream()));**

**System.out.println("[ OK ] Cliente conectado desde: "+cl.getInetAddress()+" : "+cl.getPort());**

**System.out.println("[ OK ] Esperando respuesta del cliente...");**

**String respCl = br.readLine();**

**System.out.println("[ OK ] Instruccion recibida: "+ respCl);**

**switch (respCl) {**

**//Cuando recepta 1 nos indica que quiere que le enviemos productos**

**case "1":**

**ObjectOutputStream oos = new ObjectOutputStream(cl.getOutputStream());**

**for(int i=0;i<Inventario.size();i++){**

**oos.writeObject(Inventario.get(i));**

**oos.flush();**

**System.out.println("[ OK ] Objeto enviado: "+Inventario.get(i).getNombre());**

**}**

**oos.close();**

**break;**

**case "2":**

**String Darchivo = br.readLine();**

**byte[] b = new byte[1500];**

**int n,porcentaje = 0;**

**long enviados = 0;**

**System.out.println("[ OK ] Archivo solicitado: "+Darchivo);**

**for(int i=0;i<Inventario.size();i++){**

**if(Darchivo.compareTo(Inventario.get(i).getNombre())==0){**

**File fichero = Inventario.get(i).getImagen();**

**System.out.println("[ OK ] Se ha encontrado el archivo:" + fichero.getName());**

**DataOutputStream doss = new DataOutputStream(cl.getOutputStream());**

**System.out.println(fichero.getAbsolutePath());**

**DataInputStream diss = new DataInputStream(new FileInputStream(fichero.getAbsolutePath()));**

**doss.writeUTF(fichero.getName());**

**doss.flush();**

**doss.writeLong(fichero.length());**

**doss.flush();**

**while(enviados<fichero.length()){**

**n=diss.read(b);**

**doss.write(b,0,n);**

**doss.flush();**

**enviados += n;**

**porcentaje = (int)((enviados\*100)/fichero.length());**

**System.out.print("[ OK ] Se ha enviado el "+porcentaje+" %");**

**}**

**System.out.println("[ OK ] Se ha completado envio del archivo");**

**diss.close();**

**doss.close();**

**}**

**}**

**break;**

**case "3":**

**try {**

**System.out.println("[ OK ] Recibiendo productos que desea comprar el cliente...");**

**ObjectInputStream ois = new ObjectInputStream(cl.getInputStream());**

**clienteCarritoLista = (ArrayList < Carrito >)ois.readObject();**

**for(int i=0;i<clienteCarritoLista.size();i++){**

**for(int j=0;j<5;j++){**

**if(clienteCarritoLista.get(i).getProducto() == Inventario.get(j).getID()){**

**//Significa que existe**

**int cantidad = clienteCarritoLista.get(i).getCantidad();**

**int existencias = Inventario.get(j).getExistencias();**

**Inventario.get(j).setExistencias(existencias-cantidad);**

**System.out.println("[ OK ] Inventario Actualizado con exito...");**

**j=4;**

**}**

**}**

**}**

**ois.close();**

**cl.close();**

**} catch (IOException | ClassNotFoundException e) {**

**e.printStackTrace();**

**}**

**break;**

**}**

**}**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**}**

**}**

# Conclusiones

Con el desarrollo de esta práctica aprendimos a manejar el envió de objetos en java utilizando una arquitectura cliente-servidor y además utilizamos parte del código de la práctica anterior para el envío de archivos (imágenes) del servidor al cliente.