**Actividad**

**UF2-1. Servidor socket para consulta de inventario frutería**

**Ivan Nuñez Rodriguez**

2º DAM

Programación de Procesos y Servicios

**Índice**

1. Código.
2. Resultado.
3. Enlace de Github

**Código**

**CLIENTE**

**Cliente**

import java.io.\*;  
import java.net.\*;  
import java.util.Scanner;  
  
public class Cliente {  
 private static final String *SERVIDOR* = "localhost";  
 private static final int *PUERTO* = 2001;  
  
 public static void main(String[] args) {  
 System.*out*.println(" APLICACIÓN CLIENTE");  
 System.*out*.println("-----------------------------------");  
 Scanner lector = new Scanner(System.*in*);  
 try (Socket cliente = new Socket()) {  
 InetSocketAddress direccionServidor = new InetSocketAddress(*SERVIDOR*, *PUERTO*);  
 System.*out*.println("Esperando a que el servidor acepte la conexión");  
 cliente.connect(direccionServidor);  
 System.*out*.println("Comunicación establecida con el servidor");  
 InputStream entrada = cliente.getInputStream();  
 OutputStream salida = cliente.getOutputStream();  
 String codigo;  
 System.*out*.println("Ingrese el codigo o FIN para salir");  
 while (true) {  
 System.*out*.print("Código de producto: ");  
 codigo = lector.nextLine().toUpperCase();  
 if (codigo.equalsIgnoreCase("FIN")){  
 System.*out*.println("---------Sesion finalizada---------"); break;}  
  
 salida.write((codigo + "\n").getBytes());  
 byte[] mensaje = new byte[100];  
 entrada.read(mensaje);  
 System.*out*.println("Servidor responde: " + new String(mensaje).trim());  
 }  
 } catch (IOException e) {  
 System.*out*.println("Error de comunicación");  
 System.*out*.println(e.getMessage());  
 }  
 }  
}

**SERVIDOR**

**Fruta**

import lombok.AllArgsConstructor;  
import lombok.Data;  
import java.io.Serializable;  
  
@Data  
@AllArgsConstructor  
public class Fruta implements Serializable {  
 private String nombre;  
 private int stock;  
 private float precio;  
}

**HiloCLiente**

import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStreamReader;  
import java.io.PrintWriter;  
import java.net.Socket;  
import java.util.TreeMap;  
  
class HiloCliente extends Thread {  
 private Socket socket;  
 static TreeMap<String, Fruta> *productos* = new TreeMap<>();  
 public HiloCliente(Socket socket) {  
 this.socket = socket;  
 }  
 public static Fruta buscarProducto(String codigo) {  
 return *productos*.get(codigo);  
 }  
 @Override  
 public void run() {  
 System.*out*.println("Estableciendo comunicación con " + this.getName());  
 try (BufferedReader entrada = new BufferedReader(new InputStreamReader(socket.getInputStream()));  
 PrintWriter salida = new PrintWriter(socket.getOutputStream(), true)) {  
  
 String codigo;  
 while ((codigo = entrada.readLine()) != null) {  
 Fruta producto = *buscarProducto*(codigo);  
 if (producto != null) {  
 salida.println("Producto encontrado: " + producto);  
 } else {  
 salida.println("Producto no encontrado para código: " + codigo);  
 }  
 }  
 } catch (IOException e) {  
 System.*out*.println(e.getMessage());  
 } finally {  
 try {  
 socket.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}

**ServidorFruteria**

import java.io.\*;  
import java.net.\*;  
import java.util.TreeMap;  
  
public class ServidorFruteria {  
 private static final int *PUERTO* = 2001;  
 private static final String *IP* = "localhost";  
  
  
 public static void main(String[] args) {  
 System.*out*.println("APLICACIÓN DE SERVIDOR MULTITAREA");  
 System.*out*.println("----------------------------------");  
 *cargarProductos*();  
 try (ServerSocket servidor = new ServerSocket()) {  
 InetSocketAddress direccion = new InetSocketAddress(*IP*, *PUERTO*);  
 servidor.bind(direccion);  
 System.*out*.println("Servidor listo para aceptar solicitudes");  
 System.*out*.println("Dirección IP: " + direccion.getAddress());  
 while (true) {  
 Socket cliente = servidor.accept();  
 System.*out*.println("Comunicación entrante");  
 new HiloCliente(cliente).start();  
 }  
 } catch (IOException e) {  
 System.*out*.println(e.getMessage());  
 }  
 }  
  
 private static void cargarProductos() {  
 HiloCliente.*productos*.put("PL", new Fruta("Peras limoneras", 14, 5f));  
 HiloCliente.*productos*.put("PC", new Fruta("Peras conferencia", 12, 7f));  
 HiloCliente.*productos*.put("PN", new Fruta("Plátano canario", 5, 2.5f));  
 HiloCliente.*productos*.put("BN", new Fruta("Bananas", 7, 1.3f));  
 HiloCliente.*productos*.put("TP", new Fruta("Tomates tipo pera", 8, 1.7f));  
 HiloCliente.*productos*.put("TR", new Fruta("Tomates Raf", 7, 5.3f));  
 HiloCliente.*productos*.put("UN", new Fruta("Uvas negras", 8, 3.2f));  
 HiloCliente.*productos*.put("UB", new Fruta("Uvas blancas", 5, 2.7f));  
 HiloCliente.*productos*.put("PT", new Fruta("Picotas", 8, 4.3f));  
 HiloCliente.*productos*.put("CR", new Fruta("Ciruelas rojas", 10, 2.8f));  
 HiloCliente.*productos*.put("MR", new Fruta("Melocotones rojos", 3, 2.5f));  
 HiloCliente.*productos*.put("MA", new Fruta("Melocotones amarillos", 4, 3.2f));  
 }  
  
  
}

**Ejecución**

**Servidor**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Cliente**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Enalce a Github**

<https://github.com/Ivannunezrodriguez/Programacion_de_Servicios_y_Procesos_24-25/tree/d2c63453b605c9b149ade9731105e2200988ed0f/Actividades/tema_2/Actividad%20UF2-1>