



INSTITUTO POLITECNICO NACIONAL “IPN”



ESCUELA SUPERIOR DE COMPUTO “ESCOM”

LANDEROS CORTES PEDRO JONAS

SISTEMAS DISTRIBUIDOS

TAREA 7

2023630402

Indice

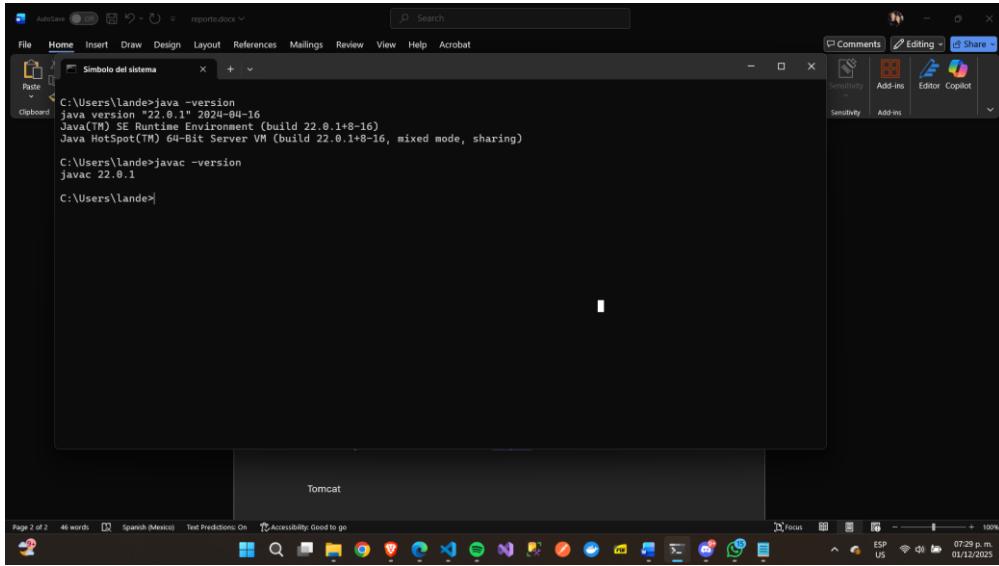
Desarrollo local	2
Instalación/Verificación de herramientas	3
Java	3
Tomcat	5
Creación de tablas en la base de datos (local)	6
Creación de usuario.....	9
Desarrollo backend.....	9
Requerimientos funcionales.	9
Compilacion	20
Pruebas en postman del backend (local)	21
Configuracion de maquina virtual	27
Reglas de entrada.....	28
Configuracion de la base de datos	29
Tablas antes.....	29
Levantar servicio	31
Probar endpoints	35
Pruebas del front	40
Conclusiones	55

Desarrollo local.

Para facilitar el desarrollo de la tarea, primero se hizo un desarrollo local de todo lo necesario para emular el contenido de la máquina virtual.

Instalación/Verificación de herramientas.

Java



Nota, para evitar incompatibilidad se hizo uso de java 8 a través de los siguientes comandos:

```
# =====
# INSTALAR JDK 8 (Eclipse Temurin) EN WINDOWS
# AUTOMÁTICAMENTE (PowerShell)
# =====
```

```
Write-Host "Descargando instalador JDK 8 Temurin..." -ForegroundColor Cyan
```

```
# URL oficial del instalador MSI (64 bits, HotSpot)
$jdkUrl = "https://github.com/adoptium/temurin8-
binaries/releases/download/jdk8u422-b05/OpenJDK8U-
jdk_x64_windows_hotspot_8u422b05.msi"
```

```
# Ruta donde se descargará
$installer = "$env:TEMP\jdk8.msi"
```

```
Invoke-WebRequest -Uri $jdkUrl -OutFile $installer
```

```
Write-Host "Instalando JDK 8..." -ForegroundColor Cyan
```

```
# Instalar JDK 8 silenciosamente
```

```
Start-Process msieexec.exe -ArgumentList "/i `"$installer`" /qn" -Wait
```

```
Write-Host "JDK 8 instalado. Configurando JAVA_HOME..." -ForegroundColor Cyan
```

```
# Buscar automáticamente la carpeta donde se instaló
```

```
$jdkPath = Get-ChildItem "C:\Program Files\Eclipse Adoptium\" | Where-Object {  
  $_.Name -like "jdk-8*" } | Select-Object -First 1
```

```
if ($jdkPath -eq $null) {
```

```
  Write-Host "ERROR: No se pudo encontrar la instalación de JDK 8." -  
  ForegroundColor Red
```

```
  exit 1
```

```
}
```

```
$fullPath = $jdkPath.FullName
```

```
# Configurar JAVA_HOME (solo para esta sesión)
```

```
$env:JAVA_HOME = $fullPath
```

```
$env:Path = "$env:JAVA_HOME\bin;$env:Path"
```

```
Write-Host "JAVA_HOME configurado a: $fullPath" -ForegroundColor Green
```

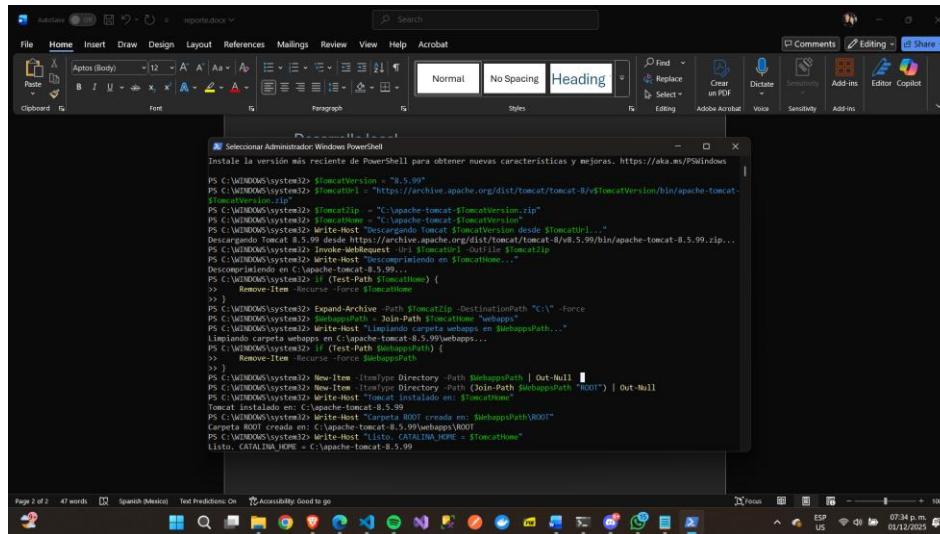
Write-Host "Verificando versión de Java..." -ForegroundColor Cyan

```
java -version
```

```
javac -version
```

Write-Host "Si ves '1.8.0_xxx', JDK 8 está instalado correctamente." -ForegroundColor Green

Tomcat



A screenshot of a Microsoft Word document window titled "Seleccionador Administrador Windows PowerShell". The document contains a PowerShell script for installing Apache Tomcat 8.5.99. The script uses the Invoke-WebRequest cmdlet to download the Tomcat zip file from the Apache website, then extracts it to a specified directory. It also creates a new item type "WebAppPath" and sets the CATALINA_HOME environment variable. The script concludes with a success message and the value of CATALINA_HOME.

```
PS C:\Windows\system2> $tomcatVersion = "8.5.99"
PS C:\Windows\system2> $tomcatUrl = "https://archive.apache.org/dist/tomcat/tomcat-8/v8TomcatVersion/bin/apache-tomcat-8.5.99.zip"
PS C:\Windows\system2> $tomcatZip = "C\apache-tomcat-$tomcatVersion.zip"
PS C:\Windows\system2> $tomcatHome = "C\apache-tomcat-$tomcatVersion"
PS C:\Windows\system2> Write-Host "Descargando Tomcat 8.5.99 en https://archive.apache.org/dist/tomcat/tomcat-8/v8.5.99/bin/apache-tomcat-8.5.99.zip..."
PS C:\Windows\system2> Invoke-WebRequest -Uri $tomcatUrl -Outfile $tomcatZip
PS C:\Windows\system2> Write-Host "Descomprimiendo en $tomcatHome..."
Extractando al directorio C\apache-tomcat-8.5.99...
PS C:\Windows\system2> Remove-Item -Recurse $tomcatHome
>>
>> Remove-Item -Recurse -Force $tomcatHome
>>
>> Remove-Item -Recurse -Force $tomcatHome
PS C:\Windows\system2> Expand-Archive -Path $tomcatZip -DestinationPath "C\\" -Force
PS C:\Windows\system2> $webappPath = Join-Path $tomcatHome "webapps"
PS C:\Windows\system2> Write-Host "Creando carpeta webapps dentro de $webappPath..."
Listo: carpeta webapps en C\apache-tomcat-8.5.99\webapps...
PS C:\Windows\system2> If (Test-Path $webappPath) {
    >> Remove-Item -Recurse $webappPath
    >>
    >> New-Item -ItemType Directory -Path $webappPath | Out-Null
    PS C:\Windows\system2> New-Item -ItemType Directory -Path (Join-Path $webappPath "ROOT") | Out-Null
    PS C:\Windows\system2> Write-Host "Carpeta ROOT creada en: $webappPath\ROOT"
    Tomcat instalado en: C\apache-tomcat-8.5.99
    PS C:\Windows\system2> Write-Host "Carpeta ROOT creada en: $webappPath\ROOT"
    Carpeta ROOT creada en: C\apache-tomcat-8.5.99\webapps\ROOT
    PS C:\Windows\system2> Write-Host "Lista: CATALINA_HOME = $tomcatHome"
    Listo: CATALINA_HOME = C\apache-tomcat-8.5.99
```

Creación de tablas en la base de datos (local)

Comandos utilizados:

CREATE DATABASE IF NOT EXISTS servicio_web

DEFAULT CHARACTER SET utf8mb4

DEFAULT COLLATE utf8mb4_general_ci;

```
USE servicio_web;
```

```
CREATE TABLE IF NOT EXISTS usuarios (
```

```
id_usuario      INT AUTO_INCREMENT PRIMARY KEY,  
email          VARCHAR(100) NOT NULL,  
nombre         VARCHAR(100) NOT NULL,  
apellido_paterno VARCHAR(100) NOT NULL,  
apellido_materno VARCHAR(100),  
fecha_nacimiento DATETIME NOT NULL,  
telefono        BIGINT,
```

```
genero      CHAR(1),
password    VARCHAR(64) NOT NULL,
token       VARCHAR(20) NULL
);
```

```
CREATE UNIQUE INDEX usuarios_1 ON usuarios(email);
```

```
CREATE TABLE IF NOT EXISTS fotos_usuarios (
    id_foto  INT AUTO_INCREMENT PRIMARY KEY,
    foto     LONGBLOB,
    id_usuario INT NOT NULL
);
```

```
ALTER TABLE fotos_usuarios
ADD CONSTRAINT fk_fotos_usuarios_usuarios
FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario);
```

```
CREATE TABLE IF NOT EXISTS stock (
    id_articulo INT AUTO_INCREMENT PRIMARY KEY,
    nombre      VARCHAR(255) NOT NULL,
    descripcion TEXT,
    precio      DECIMAL(10,2) NOT NULL,
    cantidad    INT NOT NULL
);
```

```
CREATE TABLE IF NOT EXISTS fotos_articulos (
```

```

    id_foto  INT AUTO_INCREMENT PRIMARY KEY,
    foto      LONGBLOB,
    id_articulo INT NOT NULL
);

```

```

ALTER TABLE fotos_articulos
ADD CONSTRAINT fk_fotos_articulos_stock
FOREIGN KEY (id_articulo) REFERENCES stock(id_articulo);

```

```

CREATE TABLE IF NOT EXISTS carrito_compra (
    id_usuario INT NOT NULL,
    id_articulo INT NOT NULL,
    cantidad  INT NOT NULL,
    PRIMARY KEY (id_usuario, id_articulo)
);

```

The screenshot shows the MySQL Workbench interface with the following details:

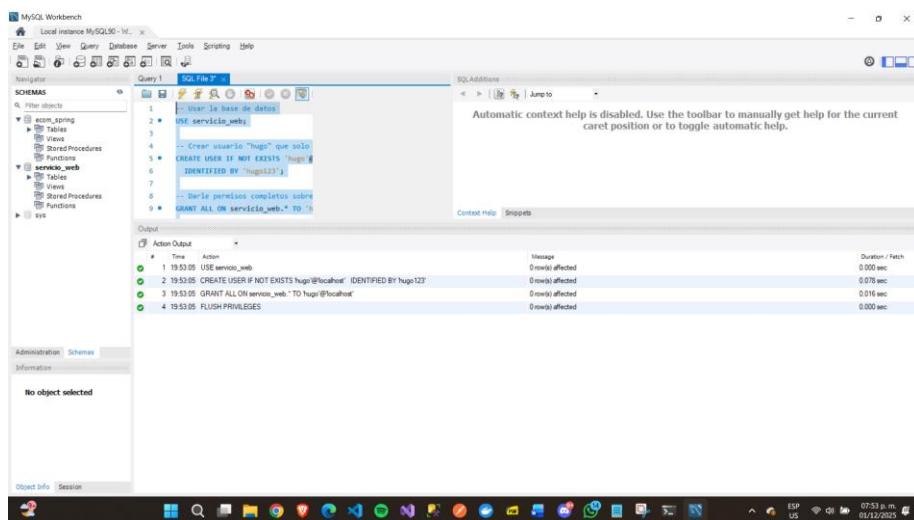
- Schemas:** Local instance MySQL80 - InnoDB
- Query Editor:** Contains the SQL code for creating the fotos_articulos table and adding a foreign key constraint.
- Output Window:** Shows the execution log with 10 actions, each with a timestamp, action, message, and duration. The log includes the creation of the database, tables, and constraints.
- Status Bar:** Displays the date and time as 01/12/2025 07:48 p.m.

```

Action Output
#  Time   Action
1  19:47:06 CREATE DATABASE IF NOT EXISTS servicio_web DEFAULT CHARACTER SET utf8mb4 DEFAULT COLLATE ...
2  19:47:06 USE servicio_web
3  19:47:06 CREATE TABLE IF NOT EXISTS usuarios ( id_usuario INT AUTO_INCREMENT PRIMARY KEY, email ...
4  19:47:06 CREATE UNIQUE INDEX usuarios_email ON usuarios(email)
5  19:47:06 CREATE TABLE IF NOT EXISTS fotos_articulos ( id_foto INT AUTO_INCREMENT PRIMARY KEY, foto ...
6  19:47:06 ALTER TABLE fotos_articulos ADD CONSTRAINT fk_fotos_articulos_stock FOREIGN KEY (id_usuario) RE ...
7  19:47:06 CREATE TABLE IF NOT EXISTS carrito_compra ( id_usuario INT NOT NULL, id_articulo INT NOT NULL, ...
8  19:47:06 CREATE TABLE IF NOT EXISTS fotos_articulos ( id_foto INT AUTO_INCREMENT PRIMARY KEY, foto ...
9  19:47:06 ALTER TABLE fotos_articulos ADD CONSTRAINT fk_fotos_articulos_stock FOREIGN KEY (id_articulo) REFE ...
10 19:47:06 CREATE TABLE IF NOT EXISTS carrito_compra ( id_usuario INT NOT NULL, id_articulo INT NOT NULL, ...

```

Creación de usuario

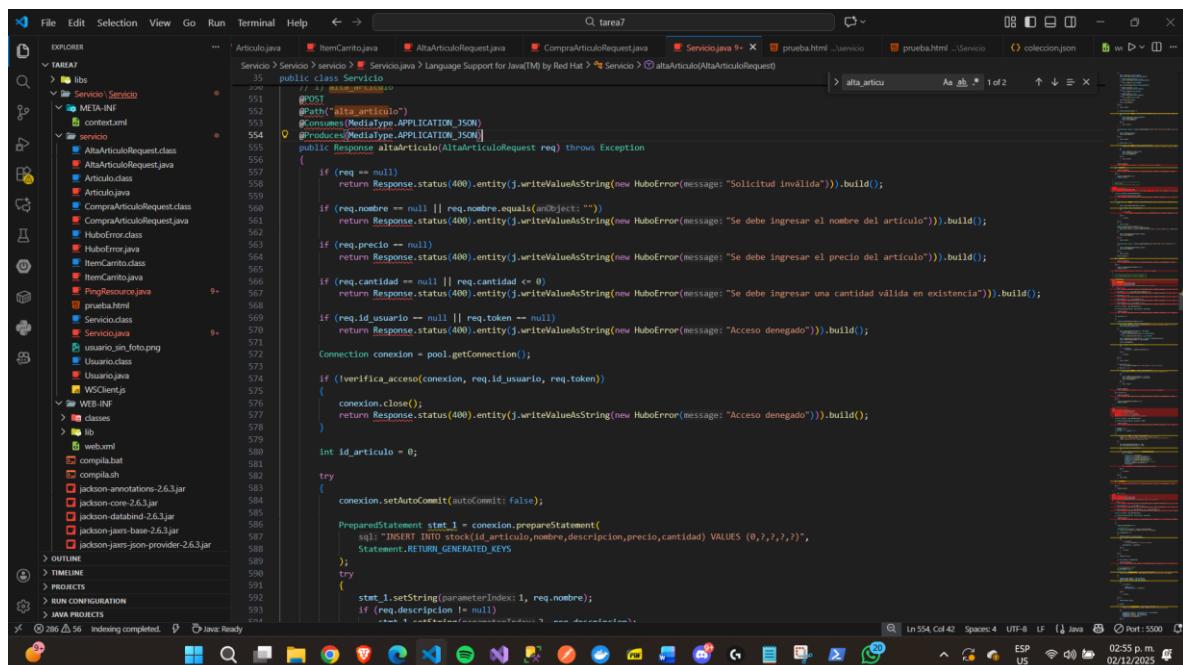


```
CREATE USER IF NOT EXISTS 'hugo'@'localhost' IDENTIFIED BY 'hugo123';
GRANT ALL ON servicio_web.* TO 'hugo'@'localhost';
```

Desarrollo backend

Requerimientos funcionales.

1. Alta articulo



```
public Response altaArticulo(AltaArticuloRequest req) throws Exception
{
    if (req == null)
        return Response.status(400).entity(j.writeValueAsString(new HuboError(message: "Solicitud inválida"))).build();

    if (req.nombre == null || req.nombre.equals("")){
        return Response.status(400).entity(j.writeValueAsString(new HuboError(message: "Se debe ingresar el nombre del artículo"))).build();
    }

    if (req.precio == null)
        return Response.status(400).entity(j.writeValueAsString(new HuboError(message: "Se debe ingresar el precio del artículo"))).build();

    if (req.cantidad == null || req.cantidad <= 0)
        return Response.status(400).entity(j.writeValueAsString(new HuboError(message: "Se debe ingresar una cantidad válida en existencia"))).build();

    if (req.id_usuario == null || req.token == null)
        return Response.status(400).entity(j.writeValueAsString(new HuboError(message: "Acceso denegado"))).build();

    Connection conexion = pool.getConnection();

    if (!verifica_acceso(conexion, req.id_usuario, req.token))
    {
        conexion.close();
        return Response.status(400).entity(j.writeValueAsString(new HuboError(message: "Acceso denegado"))).build();
    }

    int id_articulo = 0;
    try
    {
        conexion.setAutoCommit(false);

        PreparedStatement stmt_1 = conexion.prepareStatement(
            "INSERT INTO stock(id_articulo,nombre,descripcion,precio,cantidad) VALUES (0,?, ?, ?, ?)");
        stmt_1.setString(parameterIndex: 1, req.nombre);
        if (req.descripcion != null)
            stmt_1.setString(parameterIndex: 2, req.descripcion);
        if (req.precio != null)
            stmt_1.setDouble(parameterIndex: 3, req.precio);
        if (req.cantidad != null)
            stmt_1.setInt(parameterIndex: 4, req.cantidad);
        stmt_1.executeUpdate();
        id_articulo = stmt_1.getGeneratedKeys().nextInt();
        stmt_1.close();
    }
    catch (Exception e)
    {
        System.out.println(e.getMessage());
    }
}
```

```
public Response altaArticulo(AltaArticuloRequest req) throws Exception {
    connexion.setAutoCommit(false);
    PreparedStatement stmt_1 = connexion.prepareStatement(
        sql: "INSERT INTO stock(id_articulo,nombre,descripcion,precio,cantidad) VALUES (?, ?, ?, ?, ?)",
        Statement.RETURN_GENERATED_KEYS
    );
    try {
        stmt_1.setString(parameterIndex: 1, req.nombre);
        if (req.descripcion != null)
            stmt_1.setString(parameterIndex: 2, req.descripcion);
        else
            stmt_1.setNull(parameterIndex: 2, Types.VARCHAR);

        stmt_1.setBigDecimal(parameterIndex: 3, BigDecimal.valueOf(req.precio));
        stmt_1.setInt(parameterIndex: 4, req.cantidad);

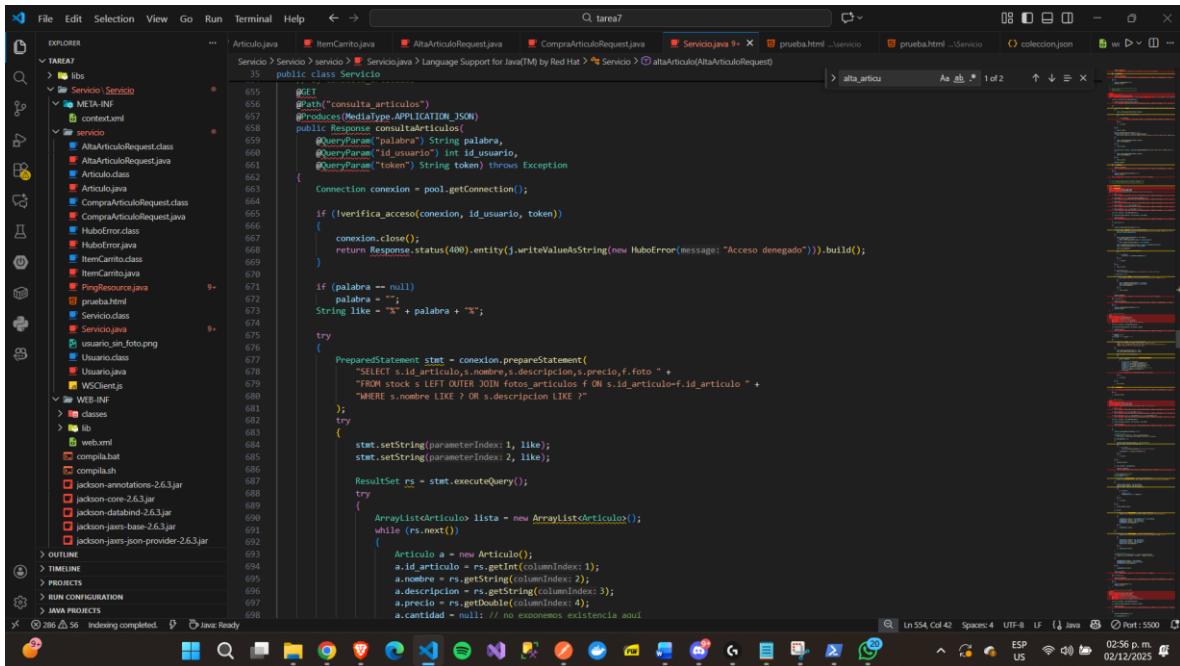
        stmt_1.executeUpdate();
        ResultSet rs = stmt_1.getGeneratedKeys();
        try {
            if (rs.next())
                id_articulo = rs.getInt(columnIndex: 1);
        } finally {
            rs.close();
        }
    } finally {
        stmt_1.close();
    }
    if (id_articulo == 0)
        throw new Exception(message: "No se pudo obtener el ID del articulo");
    if (req.foto != null) {
        PreparedStatement stmt_2 = connexion.prepareStatement(
            sql: "INSERT INTO fotos_articulos(id_foto,foto,id_articulo) VALUES (?, ?, ?)"
        );
        try {
            stmt_2.setBytes(parameterIndex: 1, req.foto);
            stmt_2.setInt(parameterIndex: 2, id_articulo);
            stmt_2.executeUpdate();
        } finally {
            stmt_2.close();
        }
        connexion.commit();
        return Response.ok("(\\"mensaje\\":\"Se dio de alta el articulo\")").build();
    }
    catch (Exception e) {
        connexion.rollback();
        return Response.status(400).entity(j.writeValueAsString(new HuboError(e.getMessage()))).build();
    }
    finally {
        connexion.setAutoCommit(autoCommit: true);
        connexion.close();
    }
}
```

```
public Response altaArticulo(AltaArticuloRequest req) throws Exception {
    connexion.setAutoCommit(false);
    PreparedStatement stmt_1 = connexion.prepareStatement(
        sql: "INSERT INTO stock(id_articulo,nombre,descripcion,precio,cantidad) VALUES (?, ?, ?, ?, ?)",
        Statement.RETURN_GENERATED_KEYS
    );
    try {
        stmt_1.setString(parameterIndex: 1, req.nombre);
        if (req.descripcion != null)
            stmt_1.setString(parameterIndex: 2, req.descripcion);
        else
            stmt_1.setNull(parameterIndex: 2, Types.VARCHAR);

        stmt_1.setBigDecimal(parameterIndex: 3, BigDecimal.valueOf(req.precio));
        stmt_1.setInt(parameterIndex: 4, req.cantidad);

        stmt_1.executeUpdate();
        ResultSet rs = stmt_1.getGeneratedKeys();
        try {
            if (rs.next())
                id_articulo = rs.getInt(columnIndex: 1);
        } finally {
            rs.close();
        }
    } finally {
        stmt_1.close();
    }
    if (id_articulo == 0)
        throw new Exception(message: "No se pudo obtener el ID del articulo");
    if (req.foto != null) {
        PreparedStatement stmt_2 = connexion.prepareStatement(
            sql: "INSERT INTO fotos_articulos(id_foto,foto,id_articulo) VALUES (?, ?, ?)"
        );
        try {
            stmt_2.setBytes(parameterIndex: 1, req.foto);
            stmt_2.setInt(parameterIndex: 2, id_articulo);
            stmt_2.executeUpdate();
        } finally {
            stmt_2.close();
        }
        connexion.commit();
        return Response.ok("(\\"mensaje\\":\"Se dio de alta el articulo\")").build();
    }
    catch (Exception e) {
        connexion.rollback();
        return Response.status(400).entity(j.writeValueAsString(new HuboError(e.getMessage()))).build();
    }
    finally {
        connexion.setAutoCommit(autoCommit: true);
        connexion.close();
    }
}
```

2. Consultar artículos



```

    ...
    @GET
    @Path("consulta_articulos")
    @Produces(MediaType.APPLICATION_JSON)
    public Response consultaArticulos(
        @QueryParam("palabra") String palabra,
        @QueryParam("id_usuario") int id_usuario,
        @QueryParam("token") String token) throws Exception
    {
        Connection conexion = pool.getConnection();

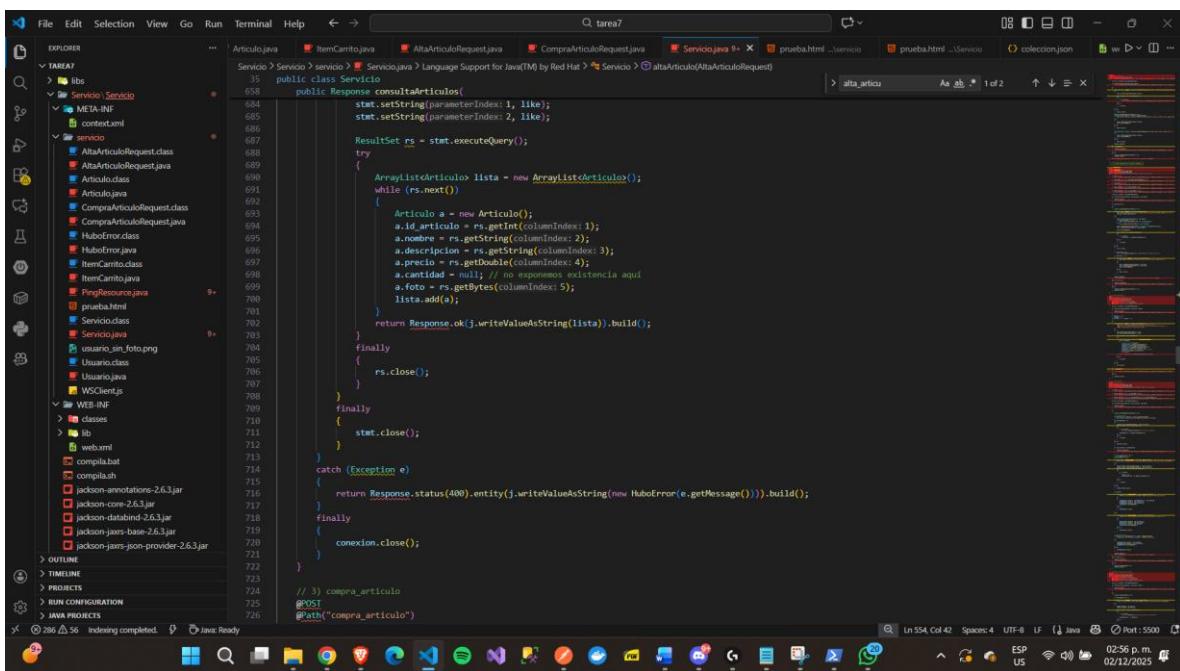
        if (!verifica_acceso(conexion, id_usuario, token))
        {
            conexion.close();
            return Response.status(400).entity(j.writeValueAsString(new HuboError(message: "Acceso denegado"))).build();
        }

        if (palabra == null)
            palabra = "%";
        String like = "%" + palabra + "%";

        try
        {
            PreparedStatement stmt = conexion.prepareStatement(
                "SELECT s.id_articulo,s.nombre,s.descripcion,s.precio,f.foto " +
                "FROM stock s LEFT OUTER JOIN fotos_articulos f ON s.id_articulo=f.id_articulo " +
                "WHERE s.nombre LIKE ? OR s.descripcion LIKE ?"
            );
            stmt.setString(parameterIndex: 1, like);
            stmt.setString(parameterIndex: 2, like);

            ResultSet rs = stmt.executeQuery();
            try
            {
                ArrayList<Articulo> lista = new ArrayList<Articulo>();
                while (rs.next())
                {
                    Articulo a = new Articulo();
                    a.id_articulo = rs.getInt(columnIndex: 1);
                    a.nombre = rs.getString(columnIndex: 2);
                    a.descripcion = rs.getString(columnIndex: 3);
                    a.precio = rs.getDouble(columnIndex: 4);
                    a.cantidad = null; // no exponemos existencia aqui
                    a.foto = rs.getBytes(columnIndex: 5);
                    lista.add(a);
                }
                return Response.ok(j.writeValueAsString(lista)).build();
            }
            finally
            {
                rs.close();
            }
        }
        finally
        {
            stmt.close();
        }
    }
    ...
}

```



```

    ...
    @POST
    @Path("compra_articulo")
    public Response compra_articulo()
    {
        ...
    }
    ...
}

```

3. Comprar artículos

```
File Edit Selection View Go Run Terminal Help < > Q, tarea? Articulo.java ItemCarrito.java AltaArticuloRequest.java CompraArticuloRequest.java Servicio.java < x prueba.html < Service prueba.html < Service colección.json < w < d ... EXPLORER ... Articulo.java ItemCarrito.java AltaArticuloRequest.java CompraArticuloRequest.java Servicio.java < x prueba.html < Service prueba.html < Service colección.json < w < d ... SERVICIOS ... Articulo.java ItemCarrito.java AltaArticuloRequest.java CompraArticuloRequest.java Servicio.java < x prueba.html < Service prueba.html < Service colección.json < w < d ... META-INF ... context.xml ... servicio ... AltaArticuloRequest.class AltaArticuloRequest.java Articulo.class Articulo.java CompraArticuloRequest.class CompraArticuloRequest.java HuboException.class HuboException.java ItemCarrito.class ItemCarrito.java PingResource.java prueba.html Servicio.class Usuario.class Usuario.java WSClient.java ... WEB-INF ... classes ... lib ... web.xml compila.bat compilash jackson-annotations-2.6.3.jar jackson-core-2.8.3.jar jackson-databind-2.6.3.jar jackson-datastore-2.6.3.jar jackson-pars-base-2.6.3.jar jackson-pars-json-provider-2.6.3.jar OUTLINE ... TIMELINE ... PROJECTS ... RUN CONFIGURATION ... JAVA PROJECTS 286 56 Indexing completed. Java Ready Articulo.java 722 } 723 } // } compra_articulo 724 @Path("compra_articulo") 725 @Consumes(MediaType.APPLICATION_JSON) 726 @Produces(MediaType.APPLICATION_JSON) 727 public Response compraArticulo(CompraArticuloRequest req) throws Exception 728 { 729 if (req == null) 730 return Response.status(400).entity(j.writeValueAsString(new HuboError(message: "Solicitud invalida"))).build(); 731 if (req.id_usuario == null || req.token == null) 732 return Response.status(400).entity(j.writeValueAsString(new HuboError(message: "Acceso denegado"))).build(); 733 if (req.id_articulo == null || req.cantidad == null || req.cantidad < 0) 734 return Response.status(400).entity(j.writeValueAsString(new HuboError(message: "Cantidad invalida"))).build(); 735 Connection conexion = pool.getConnection(); 736 737 if (!verifica_acceso(conexion, req.id_usuario, req.token)) 738 { 739 conexion.close(); 740 return Response.status(400).entity(j.writeValueAsString(new HuboError(message: "Acceso denegado"))).build(); 741 } 742 try 743 { 744 conexion.setAutoCommit(false); 745 // Consultar stock 746 PreparedStatement statStock = conexion.prepareStatement( 747 sql: "SELECT cantidad FROM stock WHERE id_articulo=? FOR UPDATE" 748 ); 749 int cantidadActual = 0; 750 try 751 { 752 statStock.setInt(parameterIndex: 1, req.id_articulo); 753 ResultSet rs = statStock.executeQuery(); 754 try 755 { 756 if (!rs.next()) 757 throw new Exception(message: "El articulo no existe"); 758 } catch (Exception e) 759 { 760 System.out.println(e.getMessage()); 761 } 762 } catch (SQLException e) 763 { 764 System.out.println(e.getMessage()); 765 } finally 766 { 767 rs.close(); 768 statStock.close(); 769 } 770 if (cantidadActual < req.cantidad) 771 throw new Exception(message: "No hay suficiente stock"); 772 else 773 { 774 cantidadActual -= req.cantidad; 775 String cantidadFinal = cantidadActual.toString(); 776 String json = j.toJson(cantidadFinal); 777 return Response.status(200).entity(json).build(); 778 } 779 } catch (Exception e) 780 { 781 System.out.println(e.getMessage()); 782 } finally 783 { 784 conexion.commit(); 785 conexion.close(); 786 } 787 } catch (SQLException e) 788 { 789 System.out.println(e.getMessage()); 790 } finally 791 { 792 conexion.close(); 793 } 794 } catch (Exception e) 795 { 796 System.out.println(e.getMessage()); 797 } finally 798 { 799 conexion.close(); 800 } 801 } catch (Exception e) 802 { 803 System.out.println(e.getMessage()); 804 } finally 805 { 806 conexion.close(); 807 } 808 } catch (Exception e) 809 { 810 System.out.println(e.getMessage()); 811 } finally 812 { 813 conexion.close(); 814 } 815 } catch (Exception e) 816 { 817 System.out.println(e.getMessage()); 818 } finally 819 { 820 conexion.close(); 821 } 822 } catch (Exception e) 823 { 824 System.out.println(e.getMessage()); 825 } finally 826 { 827 conexion.close(); 828 } 829 } catch (Exception e) 830 { 831 System.out.println(e.getMessage()); 832 } finally 833 { 834 conexion.close(); 835 } 836 } catch (Exception e) 837 { 838 System.out.println(e.getMessage()); 839 } finally 840 { 841 conexion.close(); 842 } 843 } catch (Exception e) 844 { 845 System.out.println(e.getMessage()); 846 } finally 847 { 848 conexion.close(); 849 } 850 } catch (Exception e) 851 { 852 System.out.println(e.getMessage()); 853 } finally 854 { 855 conexion.close(); 856 } 857 } catch (Exception e) 858 { 859 System.out.println(e.getMessage()); 860 } finally 861 { 862 conexion.close(); 863 } 864 } catch (Exception e) 865 { 866 System.out.println(e.getMessage()); 867 } finally 868 { 869 conexion.close(); 870 } 871 } catch (Exception e) 872 { 873 System.out.println(e.getMessage()); 874 } finally 875 { 876 conexion.close(); 877 } 878 } catch (Exception e) 879 { 880 System.out.println(e.getMessage()); 881 } finally 882 { 883 conexion.close(); 884 } 885 } catch (Exception e) 886 { 887 System.out.println(e.getMessage()); 888 } finally 889 { 890 conexion.close(); 891 } 892 } catch (Exception e) 893 { 894 System.out.println(e.getMessage()); 895 } finally 896 { 897 conexion.close(); 898 } 899 } catch (Exception e) 900 { 901 System.out.println(e.getMessage()); 902 } finally 903 { 904 conexion.close(); 905 } 906 } catch (Exception e) 907 { 908 System.out.println(e.getMessage()); 909 } finally 910 { 911 conexion.close(); 912 } 913 } catch (Exception e) 914 { 915 System.out.println(e.getMessage()); 916 } finally 917 { 918 conexion.close(); 919 } 920 } catch (Exception e) 921 { 922 System.out.println(e.getMessage()); 923 } finally 924 { 925 conexion.close(); 926 } 927 } catch (Exception e) 928 { 929 System.out.println(e.getMessage()); 930 } finally 931 { 932 conexion.close(); 933 } 934 } catch (Exception e) 935 { 936 System.out.println(e.getMessage()); 937 } finally 938 { 939 conexion.close(); 940 } 941 } catch (Exception e) 942 { 943 System.out.println(e.getMessage()); 944 } finally 945 { 946 conexion.close(); 947 } 948 } catch (Exception e) 949 { 950 System.out.println(e.getMessage()); 951 } finally 952 { 953 conexion.close(); 954 } 955 } catch (Exception e) 956 { 957 System.out.println(e.getMessage()); 958 } finally 959 { 960 conexion.close(); 961 } 962 } catch (Exception e) 963 { 964 System.out.println(e.getMessage()); 965 } finally 966 { 967 conexion.close(); 968 } 969 } catch (Exception e) 970 { 971 System.out.println(e.getMessage()); 972 } finally 973 { 974 conexion.close(); 975 } 976 } catch (Exception e) 977 { 978 System.out.println(e.getMessage()); 979 } finally 980 { 981 conexion.close(); 982 } 983 } catch (Exception e) 984 { 985 System.out.println(e.getMessage()); 986 } finally 987 { 988 conexion.close(); 989 } 990 } catch (Exception e) 991 { 992 System.out.println(e.getMessage()); 993 } finally 994 { 995 conexion.close(); 996 } 997 } catch (Exception e) 998 { 999 System.out.println(e.getMessage()); 1000 } finally 1001 { 1002 conexion.close(); 1003 } 1004 } catch (Exception e) 1005 { 1006 System.out.println(e.getMessage()); 1007 } finally 1008 { 1009 conexion.close(); 1010 } 1011 } catch (Exception e) 1012 { 1013 System.out.println(e.getMessage()); 1014 } finally 1015 { 1016 conexion.close(); 1017 } 1018 } catch (Exception e) 1019 { 1020 System.out.println(e.getMessage()); 1021 } finally 1022 { 1023 conexion.close(); 1024 } 1025 } catch (Exception e) 1026 { 1027 System.out.println(e.getMessage()); 1028 } finally 1029 { 1030 conexion.close(); 1031 } 1032 } catch (Exception e) 1033 { 1034 System.out.println(e.getMessage()); 1035 } finally 1036 { 1037 conexion.close(); 1038 } 1039 } catch (Exception e) 1040 { 1041 System.out.println(e.getMessage()); 1042 } finally 1043 { 1044 conexion.close(); 1045 } 1046 } catch (Exception e) 1047 { 1048 System.out.println(e.getMessage()); 1049 } finally 1050 { 1051 conexion.close(); 1052 } 1053 } catch (Exception e) 1054 { 1055 System.out.println(e.getMessage()); 1056 } finally 1057 { 1058 conexion.close(); 1059 } 1060 } catch (Exception e) 1061 { 1062 System.out.println(e.getMessage()); 1063 } finally 1064 { 1065 conexion.close(); 1066 } 1067 } catch (Exception e) 1068 { 1069 System.out.println(e.getMessage()); 1070 } finally 1071 { 1072 conexion.close(); 1073 } 1074 } catch (Exception e) 1075 { 1076 System.out.println(e.getMessage()); 1077 } finally 1078 { 1079 conexion.close(); 1080 } 1081 } catch (Exception e) 1082 { 1083 System.out.println(e.getMessage()); 1084 } finally 1085 { 1086 conexion.close(); 1087 } 1088 } catch (Exception e) 1089 { 1090 System.out.println(e.getMessage()); 1091 } finally 1092 { 1093 conexion.close(); 1094 } 1095 } catch (Exception e) 1096 { 1097 System.out.println(e.getMessage()); 1098 } finally 1099 { 1100 conexion.close(); 1101 } 1102 } catch (Exception e) 1103 { 1104 System.out.println(e.getMessage()); 1105 } finally 1106 { 1107 conexion.close(); 1108 } 1109 } catch (Exception e) 1110 { 1111 System.out.println(e.getMessage()); 1112 } finally 1113 { 1114 conexion.close(); 1115 } 1116 } catch (Exception e) 1117 { 1118 System.out.println(e.getMessage()); 1119 } finally 1120 { 1121 conexion.close(); 1122 } 1123 } catch (Exception e) 1124 { 1125 System.out.println(e.getMessage()); 1126 } finally 1127 { 1128 conexion.close(); 1129 } 1130 } catch (Exception e) 1131 { 1132 System.out.println(e.getMessage()); 1133 } finally 1134 { 1135 conexion.close(); 1136 } 1137 } catch (Exception e) 1138 { 1139 System.out.println(e.getMessage()); 1140 } finally 1141 { 1142 conexion.close(); 1143 } 1144 } catch (Exception e) 1145 { 1146 System.out.println(e.getMessage()); 1147 } finally 1148 { 1149 conexion.close(); 1150 } 1151 } catch (Exception e) 1152 { 1153 System.out.println(e.getMessage()); 1154 } finally 1155 { 1156 conexion.close(); 1157 } 1158 } catch (Exception e) 1159 { 1160 System.out.println(e.getMessage()); 1161 } finally 1162 { 1163 conexion.close(); 1164 } 1165 } catch (Exception e) 1166 { 1167 System.out.println(e.getMessage()); 1168 } finally 1169 { 1170 conexion.close(); 1171 } 1172 } catch (Exception e) 1173 { 1174 System.out.println(e.getMessage()); 1175 } finally 1176 { 1177 conexion.close(); 1178 } 1179 } catch (Exception e) 1180 { 1181 System.out.println(e.getMessage()); 1182 } finally 1183 { 1184 conexion.close(); 1185 } 1186 } catch (Exception e) 1187 { 1188 System.out.println(e.getMessage()); 1189 } finally 1190 { 1191 conexion.close(); 1192 } 1193 } catch (Exception e) 1194 { 1195 System.out.println(e.getMessage()); 1196 } finally 1197 { 1198 conexion.close(); 1199 } 1200 } catch (Exception e) 1201 { 1202 System.out.println(e.getMessage()); 1203 } finally 1204 { 1205 conexion.close(); 1206 } 1207 } catch (Exception e) 1208 { 1209 System.out.println(e.getMessage()); 1210 } finally 1211 { 1212 conexion.close(); 1213 } 1214 } catch (Exception e) 1215 { 1216 System.out.println(e.getMessage()); 1217 } finally 1218 { 1219 conexion.close(); 1220 } 1221 } catch (Exception e) 1222 { 1223 System.out.println(e.getMessage()); 1224 } finally 1225 { 1226 conexion.close(); 1227 } 1228 } catch (Exception e) 1229 { 1230 System.out.println(e.getMessage()); 1231 } finally 1232 { 1233 conexion.close(); 1234 } 1235 } catch (Exception e) 1236 { 1237 System.out.println(e.getMessage()); 1238 } finally 1239 { 1240 conexion.close(); 1241 } 1242 } catch (Exception e) 1243 { 1244 System.out.println(e.getMessage()); 1245 } finally 1246 { 1247 conexion.close(); 1248 } 1249 } catch (Exception e) 1250 { 1251 System.out.println(e.getMessage()); 1252 } finally 1253 { 1254 conexion.close(); 1255 } 1256 } catch (Exception e) 1257 { 1258 System.out.println(e.getMessage()); 1259 } finally 1260 { 1261 conexion.close(); 1262 } 1263 } catch (Exception e) 1264 { 1265 System.out.println(e.getMessage()); 1266 } finally 1267 { 1268 conexion.close(); 1269 } 1270 } catch (Exception e) 1271 { 1272 System.out.println(e.getMessage()); 1273 } finally 1274 { 1275 conexion.close(); 1276 } 1277 } catch (Exception e) 1278 { 1279 System.out.println(e.getMessage()); 1280 } finally 1281 { 1282 conexion.close(); 1283 } 1284 } catch (Exception e) 1285 { 1286 System.out.println(e.getMessage()); 1287 } finally 1288 { 1289 conexion.close(); 1290 } 1291 } catch (Exception e) 1292 { 1293 System.out.println(e.getMessage()); 1294 } finally 1295 { 1296 conexion.close(); 1297 } 1298 } catch (Exception e) 1299 { 1300 System.out.println(e.getMessage()); 1301 } finally 1302 { 1303 conexion.close(); 1304 } 1305 } catch (Exception e) 1306 { 1307 System.out.println(e.getMessage()); 1308 } finally 1309 { 1310 conexion.close(); 1311 } 1312 } catch (Exception e) 1313 { 1314 System.out.println(e.getMessage()); 1315 } finally 1316 { 1317 conexion.close(); 1318 } 1319 } catch (Exception e) 1320 { 1321 System.out.println(e.getMessage()); 1322 } finally 1323 { 1324 conexion.close(); 1325 } 1326 } catch (Exception e) 1327 { 1328 System.out.println(e.getMessage()); 1329 } finally 1330 { 1331 conexion.close(); 1332 } 1333 } catch (Exception e) 1334 { 1335 System.out.println(e.getMessage()); 1336 } finally 1337 { 1338 conexion.close(); 1339 } 1340 } catch (Exception e) 1341 { 1342 System.out.println(e.getMessage()); 1343 } finally 1344 { 1345 conexion.close(); 1346 } 1347 } catch (Exception e) 1348 { 1349 System.out.println(e.getMessage()); 1350 } finally 1351 { 1352 conexion.close(); 1353 } 1354 } catch (Exception e) 1355 { 1356 System.out.println(e.getMessage()); 1357 } finally 1358 { 1359 conexion.close(); 1360 } 1361 } catch (Exception e) 1362 { 1363 System.out.println(e.getMessage()); 1364 } finally 1365 { 1366 conexion.close(); 1367 } 1368 } catch (Exception e) 1369 { 1370 System.out.println(e.getMessage()); 1371 } finally 1372 { 1373 conexion.close(); 1374 } 1375 } catch (Exception e) 1376 { 1377 System.out.println(e.getMessage()); 1378 } finally 1379 { 1380 conexion.close(); 1381 } 1382 } catch (Exception e) 1383 { 1384 System.out.println(e.getMessage()); 1385 } finally 1386 { 1387 conexion.close(); 1388 } 1389 } catch (Exception e) 1390 { 1391 System.out.println(e.getMessage()); 1392 } finally 1393 { 1394 conexion.close(); 1395 } 1396 } catch (Exception e) 1397 { 1398 System.out.println(e.getMessage()); 1399 } finally 1400 { 1401 conexion.close(); 1402 } 1403 } catch (Exception e) 1404 { 1405 System.out.println(e.getMessage()); 1406 } finally 1407 { 1408 conexion.close(); 1409 } 1410 } catch (Exception e) 1411 { 1412 System.out.println(e.getMessage()); 1413 } finally 1414 { 1415 conexion.close(); 1416 } 1417 } catch (Exception e) 1418 { 1419 System.out.println(e.getMessage()); 1420 } finally 1421 { 1422 conexion.close(); 1423 } 1424 } catch (Exception e) 1425 { 1426 System.out.println(e.getMessage()); 1427 } finally 1428 { 1429 conexion.close(); 1430 } 1431 } catch (Exception e) 1432 { 1433 System.out.println(e.getMessage()); 1434 } finally 1435 { 1436 conexion.close(); 1437 } 1438 } catch (Exception e) 1439 { 1440 System.out.println(e.getMessage()); 1441 } finally 1442 { 1443 conexion.close(); 1444 } 1445 } catch (Exception e) 1446 { 1447 System.out.println(e.getMessage()); 1448 } finally 1449 { 1450 conexion.close(); 1451 } 1452 } catch (Exception e) 1453 { 1454 System.out.println(e.getMessage()); 1455 } finally 1456 { 1457 conexion.close(); 1458 } 1459 } catch (Exception e) 1460 { 1461 System.out.println(e.getMessage()); 1462 } finally 1463 { 1464 conexion.close(); 1465 } 1466 } catch (Exception e) 1467 { 1468 System.out.println(e.getMessage()); 1469 } finally 1470 { 1471 conexion.close(); 1472 } 1473 } catch (Exception e) 1474 { 1475 System.out.println(e.getMessage()); 1476 } finally 1477 { 1478 conexion.close(); 1479 } 1480 } catch (Exception e) 1481 { 1482 System.out.println(e.getMessage()); 1483 } finally 1484 { 1485 conexion.close(); 1486 } 1487 } catch (Exception e) 1488 { 1489 System.out.println(e.getMessage()); 1490 } finally 1491 { 1492 conexion.close(); 1493 } 1494 } catch (Exception e) 1495 { 1496 System.out.println(e.getMessage()); 1497 } finally 1498 { 1499 conexion.close(); 1500 } 1501 } catch (Exception e) 1502 { 1503 System.out.println(e.getMessage()); 1504 } finally 1505 { 1506 conexion.close(); 1507 } 1508 } catch (Exception e) 1509 { 1510 System.out.println(e.getMessage()); 1511 } finally 1512 { 1513 conexion.close(); 1514 } 1515 } catch (Exception e) 1516 { 1517 System.out.println(e.getMessage()); 1518 } finally 1519 { 1520 conexion.close(); 1521 } 1522 } catch (Exception e) 1523 { 1524 System.out.println(e.getMessage()); 1525 } finally 1526 { 1527 conexion.close(); 1528 } 1529 } catch (Exception e) 1530 { 1531 System.out.println(e.getMessage()); 1532 } finally 1533 { 1534 conexion.close(); 1535 } 1536 } catch (Exception e) 1537 { 1538 System.out.println(e.getMessage()); 1539 } finally 1540 { 1541 conexion.close(); 1542 } 1543 } catch (Exception e) 1544 { 1545 System.out.println(e.getMessage()); 1546 } finally 1547 { 1548 conexion.close(); 1549 } 1550 } catch (Exception e) 1551 { 1552 System.out.println(e.getMessage()); 1553 } finally 1554 { 1555 conexion.close(); 1556 } 1557 } catch (Exception e) 1558 { 1559 System.out.println(e.getMessage()); 1560 } finally 1561 { 1562 conexion.close(); 1563 } 1564 } catch (Exception e) 1565 { 1566 System.out.println(e.getMessage()); 1567 } finally 1568 { 1569 conexion.close(); 1570 } 1571 } catch (Exception e) 1572 { 1573 System.out.println(e.getMessage()); 1574 } finally 1575 { 1576 conexion.close(); 1577 } 1578 } catch (Exception e) 1579 { 1580 System.out.println(e.getMessage()); 1581 } finally 1582 { 1583 conexion.close(); 1584 } 1585 } catch (Exception e) 1586 { 1587 System.out.println(e.getMessage()); 1588 } finally 1589 { 1590 conexion.close(); 1591 } 1592 } catch (Exception e) 1593 { 1594 System.out.println(e.getMessage()); 1595 } finally 1596 { 1597 conexion.close(); 1598 } 1599 } catch (Exception e) 1600 { 1601 System.out.println(e.getMessage()); 1602 } finally 1603 { 1604 conexion.close(); 1605 } 1606 } catch (Exception e) 1607 { 1608 System.out.println(e.getMessage()); 1609 } finally 1610 { 1611 conexion.close(); 1612 } 1613 } catch (Exception e) 1614 { 1615 System.out.println(e.getMessage()); 1616 } finally 1617 { 1618 conexion.close(); 1619 } 1620 } catch (Exception e) 1621 { 1622 System.out.println(e.getMessage()); 1623 } finally 1624 { 1625 conexion.close(); 1626 } 1627 } catch (Exception e) 1628 { 1629 System.out.println(e.getMessage()); 1630 } finally 1631 { 1632 conexion.close(); 1633 } 1634 } catch (Exception e) 1635 { 1636 System.out.println(e.getMessage()); 1637 } finally 1638 { 1639 conexion.close(); 1640 } 1641 } catch (Exception e) 1642 { 1643 System.out.println(e.getMessage()); 1644 } finally 1645 { 1646 conexion.close(); 1647 } 1648 } catch (Exception e) 1649 { 1650 System.out.println(e.getMessage()); 1651 } finally 1652 { 1653 conexion.close(); 1654 } 1655 } catch (Exception e) 1656 { 1657 System.out.println(e.getMessage()); 1658 } finally 1659 { 1660 conexion.close(); 1661 } 1662 } catch (Exception e) 1663 { 1664 System.out.println(e.getMessage()); 1665 } finally 1666 { 1667 conexion.close(); 1668 } 1669 } catch (Exception e) 1670 { 1671 System.out.println(e.getMessage()); 1672 } finally 1673 { 1674 conexion.close(); 1675 } 1676 } catch (Exception e) 1677 { 1678 System.out.println(e.getMessage()); 1679 } finally 1680 { 1681 conexion.close(); 1682 } 1683 } catch (Exception e) 1684 { 1685 System.out.println(e.getMessage()); 1686 } finally 1687 { 1688 conexion.close(); 1689 } 1690 } catch (Exception e) 1691 { 1692 System.out.println(e.getMessage()); 1693 } finally 1694 { 1695 conexion.close(); 1696 } 1697 } catch (Exception e) 1698 { 1699 System.out.println(e.getMessage()); 1700 } finally 1701 { 1702 conexion.close(); 1703 } 1704 } catch (Exception e) 1705 { 1706 System.out.println(e.getMessage()); 1707 } finally 1708 { 1709 conexion.close(); 1710 } 1711 } catch (Exception e) 1712 { 1713 System.out.println(e.getMessage()); 1714 } finally 1715 { 1716 conexion.close(); 1717 } 1718 } catch (Exception e) 1719 { 1720 System.out.println(e.getMessage()); 1721 } finally 1722 { 1723 conexion.close(); 1724 } 1725 } catch (Exception e) 1726 { 1727 System.out.println(e.getMessage()); 1728 } finally 1729 { 1730 conexion.close(); 1731 } 1732 } catch (Exception e) 1733 { 1734 System.out.println(e.getMessage()); 1735 } finally 1736 { 1737 conexion.close(); 1738 } 1739 } catch (Exception e) 1740 { 1741 System.out.println(e.getMessage()); 1742 } finally 1743 { 1744 conexion.close(); 1745 } 1746 } catch (Exception e) 1747 { 1748 System.out.println(e.getMessage()); 1749 } finally 1750 { 1751 conexion.close(); 1752 } 1753 } catch (Exception e) 1754 { 1755 System.out.println(e.getMessage()); 1756 } finally 1757 { 1758 conexion.close(); 1759 } 1760 } catch (Exception e) 1761 { 1762 System.out.println(e.getMessage()); 1763 } finally 1764 { 1765 conexion.close(); 1766 } 1767 } catch (Exception e) 1768 { 1769 System.out.println(e.getMessage()); 1770 } finally 1771 { 1772 conexion.close(); 1773 } 1774 } catch (Exception e) 1775 { 1776 System.out.println(e.getMessage()); 1777 } finally 1778 { 1779 conexion.close(); 1780 } 1781 } catch (Exception e) 1782 { 1783 System.out.println(e.getMessage()); 1784 } finally 1785 { 1786 conexion.close(); 1787 } 1788 } catch (Exception e) 1789 { 1790 System.out.println(e.getMessage()); 1791 } finally 1792 { 1793 conexion.close(); 1794 } 1795 } catch (Exception e) 1796 { 1797 System.out.println(e.getMessage()); 1798 } finally 1799 { 1800 conexion.close(); 1801 } 1802 } catch (Exception e) 1803 { 1804 System.out.println(e.getMessage()); 1805 } finally 1806 { 1807 conexion.close(); 1808 } 1809 } catch (Exception e) 1810 { 1811 System.out.println(e.getMessage()); 1812 } finally 1813 { 1814 conexion.close(); 1815 } 1816 } catch (Exception e) 1817 { 1818 System.out.println(e.getMessage()); 1819 } finally 1820 { 1821 conexion.close(); 1822 } 1823 } catch (Exception e) 1824 { 1825 System.out.println(e.getMessage()); 1826 } finally 1827 { 1828 conexion.close(); 1829 } 1830 } catch (Exception e) 1831 { 1832 System.out.println(e.getMessage()); 1833 } finally 1834 { 1835 conexion.close(); 1836 } 1837 } catch (Exception e) 1838 { 1839 System.out.println(e.getMessage()); 1840 } finally 1841 { 1842 conexion.close(); 1843 } 1844 } catch (Exception e) 1845 { 1846 System.out.println(e.getMessage()); 1847 } finally 1848 { 1849 conexion.close(); 1850 } 1851 } catch (Exception e) 1852 { 1853 System.out.println(e.getMessage()); 1854 } finally 1855 { 1856 conexion.close(); 1857 } 1858 } catch (Exception e) 1859 { 1860 System.out.println(e.getMessage()); 1861 } finally 1862 { 1863 conexion.close(); 1864 } 1865 } catch (Exception e) 1866 { 1867 System.out.println(e.getMessage()); 1868 } finally 1869 { 1870 conexion.close(); 1871 } 1872 } catch (Exception e) 1873 { 1874 System.out.println(e.getMessage()); 1875 } finally 1876 { 1877 conexion.close(); 1878 } 1879 } catch (Exception e) 1880 { 1881 System.out.println(e.getMessage()); 1882 } finally 1883 { 1884 conexion.close(); 1885 } 1886 } catch (Exception e) 1887 { 1888 System.out.println(e.getMessage()); 1889 } finally 1890 { 1891 conexion.close(); 1892 } 1893 } catch (Exception e) 1894 { 1895 System.out.println(e.getMessage()); 1896 } finally 1897 { 1898 conexion.close(); 1899 } 1900 } catch (Exception e) 1901 { 1902 System.out.println(e.getMessage()); 1903 } finally 1904 { 1905 conexion.close(); 1906 } 1907 } catch (Exception e) 1908 { 1909 System.out.println(e.getMessage()); 1910 } finally 1911 { 1912 conexion.close(); 1913 } 1914 } catch (Exception e) 1915 { 1916 System.out.println(e.getMessage()); 1917 } finally 1918 { 1919 conexion.close(); 1920 } 1921 } catch (Exception e) 1922 { 1923 System.out.println(e.getMessage()); 1924 } finally 1925 { 1926 conexion.close(); 1927 } 1928 } catch (Exception e) 1929 { 1930 System.out.println(e.getMessage()); 1931 } finally 1932 { 1933 conexion.close(); 1934 } 1935 } catch (Exception e) 1936 { 1937 System.out.println(e.getMessage()); 1938 } finally 1939 { 1940 conexion.close(); 1941 } 1942 } catch (Exception e) 1943 { 1944 System.out.println(e.getMessage()); 1945 } finally 1946 { 1947 conexion.close(); 1948 } 1949 } catch (Exception e) 1950 { 1951 System.out.println(e.getMessage()); 1952 } finally 1953 { 1954 conexion.close(); 1955 } 1956 } catch (Exception e) 1957 { 1958 System.out.println(e.getMessage()); 1959 } finally 1960 { 1961 conexion.close(); 1962 } 1963 } catch (Exception e) 1964 { 1965 System.out.println(e.getMessage()); 1966 } finally 1967 { 1968 conexion.close(); 1969 } 1970 } catch (Exception e) 1971 { 1972 System.out.println(e.getMessage()); 1973 } finally 1974 { 1975 conexion.close(); 1976 } 1977 } catch (Exception e) 1978 { 1979 System.out.println(e.getMessage()); 1980 } finally 1981 { 1982 conexion.close(); 1983 } 1984 } catch (Exception e) 1985 { 1986 System.out.println(e.getMessage()); 1987 } finally 1988 { 1989 conexion.close(); 1990 } 1991 } catch (Exception e) 1992 { 1993 System.out.println(e.getMessage()); 1994 } finally 1995 { 1996 conexion.close(); 1997 } 1998 } catch (Exception e) 1999 { 2000 System.out.println(e.getMessage()); 2001 } finally 2002 { 2003 conexion.close(); 2004 } 2005 } catch (Exception e) 2006 { 2007 System.out.println(e.getMessage()); 2008 } finally 2009 { 2010 conexion.close(); 2011 } 2012 } catch (Exception e) 2013 { 2014 System.out.println(e.getMessage()); 2015 } finally 2016 { 2017 conexion.close(); 2018 } 2019 } catch (Exception e) 2020 { 2021 System.out.println(e.getMessage()); 2022 } finally 2023 { 2024 conexion.close(); 2025 } 2026 } catch (Exception e) 2027 { 2028 System.out.println(e.getMessage()); 2029 } finally 2030 { 2031 conexion.close(); 2032 } 2033 } catch (Exception e) 2034 { 2035 System.out.println(e.getMessage()); 2036 } finally 2037 { 2038 conexion.close(); 2039 } 2040 } catch (Exception e) 2041 { 2042 System.out.println(e.getMessage()); 2043 } finally 2044 { 2045 conexion.close(); 2046 } 2047 } catch (Exception e) 2048 { 2049 System.out.println(e.getMessage()); 2050 } finally 2051 { 2052 conexion.close(); 2053 } 2054 } catch (Exception e) 2055 { 2056 System.out.println(e.getMessage()); 2057 } finally 2058 { 2059 conexion.close(); 2060 } 2061 } catch (Exception e) 2062 { 2063 System.out.println(e.getMessage()); 2064 } finally 2065 { 2066 conexion.close(); 2067 } 2068 } catch (Exception e) 2069 { 2070 System.out.println(e.getMessage()); 2071 } finally 2072 { 2073 conexion.close(); 2074 } 2075 } catch (Exception e) 2076 { 2077 System.out.println(e.getMessage()); 2078 } finally 2079 { 2080 conexion.close(); 2081 } 2082 } catch (Exception e) 2083 { 2084 System.out.println(e.getMessage()); 2085 } finally 2086 { 2087 conexion.close(); 2088 } 2089 } catch (Exception e) 2090 { 2091 System.out.println(e.getMessage()); 2092 } finally 2093 { 2094 conexion.close(); 2095 } 2096 } catch (Exception e) 2097 { 2098 System.out.println(e.getMessage()); 2099 } finally 2100 { 2101 conexion.close(); 2102 } 2103 } catch (Exception e) 2104 { 2105 System.out.println(e.getMessage()); 2106 } finally 2107 { 2108 conexion.close(); 2109 } 2110 } catch (Exception e) 2111 { 2112 System.out.println(e.getMessage()); 2113 } finally 2114 { 2115 conexion.close(); 2116 } 2117 } catch (Exception e) 2118 { 2119 System.out.println(e.getMessage()); 2120 } finally 2121 { 2122 conexion.close(); 2123 } 2124 } catch (Exception e) 2125 { 2126 System.out.println(e.getMessage()); 2127 } finally 2128 { 2129 conexion.close(); 2130 } 2131 } catch (Exception e) 2132 { 2133 System.out.println(e.getMessage()); 2134 } finally 2135 { 2136 conexion.close(); 2137 } 2138 } catch (Exception e) 2139 { 2140 System.out.println(e.getMessage()); 2141 } finally 2142 { 2143 conexion.close(); 2144 } 2145 } catch (Exception e) 2146 { 2147 System.out.println(e.getMessage()); 2148 } finally 2149 { 2150 conexion.close(); 2151 } 2152 } catch (Exception e) 2153 { 2154 System.out.println(e.getMessage()); 2155 } finally 2156 { 2157 conexion.close(); 2158 } 2159 } catch (Exception e) 2160 { 2161 System.out.println(e.getMessage()); 2162 } finally 2163 { 2164 conexion.close(); 2165 } 2166 } catch (Exception e) 2167 { 2168 System.out.println(e.getMessage()); 2169 } finally 2170 { 2171 conexion.close(); 2172 } 2173 } catch (Exception e) 2174 { 2175 System.out.println(e.getMessage()); 2176 } finally 2177 { 2178 conexion.close(); 2179 } 2180 } catch (Exception e) 2181 { 2182 System.out.println(e.getMessage()); 2183 } finally 2184 { 2185 conexion.close(); 2186 } 2187 } catch (Exception e) 2188 { 2189 System.out.println(e.getMessage()); 2190 } finally 2191 { 2192 conexion.close(); 2193 } 2194 } catch (Exception e) 2195 { 2196 System.out.println(e.getMessage()); 2197 } finally 2198 { 2199 conexion.close(); 2200 } 2201 } catch (Exception e) 2202 { 2203 System.out.println(e.getMessage()); 2204 } finally 2205 { 2206 conexion.close(); 2207 } 2208 } catch (Exception e) 2209 { 2210 System.out.println(e.getMessage()); 2211 } finally 2212 { 2213 conexion.close(); 2214 } 2215 } catch (Exception e) 2216 { 2217 System.out.println(e.getMessage()); 2218 } finally 2219 { 2220 conexion.close(); 2221 } 2222 } catch (Exception e) 2223 { 2224 System.out.println(e.getMessage()); 2225 } finally 2226 { 2227 conexion.close(); 2228 } 2229 } catch (Exception e) 2230 { 2231 System.out.println(e.getMessage()); 2232 } finally 2233 { 2234 conexion.close(); 2235 } 2236 } catch (Exception e) 2237 { 2238 System.out.println(e.getMessage()); 2239 } finally 2240 { 2241 conexion.close(); 2242 } 2243 } catch (Exception e) 2244
```

File Edit Selection View Go Run Terminal Help ↺ ↻

Q, tarea7

EXPLORER

TAREA7

> libs

Service | Servicio

META-INF

context.xml

service

AltArticuloRequest.class

AltArticuloRequest.java

Articulo.class

Articulo.java

CompraArticuloRequest.class

CompraArticuloRequest.java

HuevoRequest.java

HuevoRequest.class

HuevoRequest.java

ItemCarrito.class

ItemCarrito.java

PingResource.java

prueba.html

prueba.html

Servicio.class

Servicio.java

usuario_sin_foto.png

Usuario.class

Usuario.java

WSClient.js

WEB-INF

classes

libs

web.xml

compila.bat

compilacion.sh

jackson-annotations-2.6.3.jar

jackson-core-2.6.3.jar

jackson-databind-2.6.3.jar

jackson-jars-base-2.6.3.jar

jackson-jars-json-provider-2.6.3.jar

OUTLINE

TIMELINE

PROJECTS

RUN CONFIGURATION

JAVA PROJECTS

286 56 Indexing completed. Java Ready

Articulo.java

ItemCarrito.java

AltaArticuloRequest.java

CompraArticuloRequest.java

Servicio.java

prueba.html

colección.json

... Articulo.java

729 public class Servicio

887 {

888 public Response compraArticulo(CompraArticuloRequest req) throws Exception

889 {

890 finally

891 {

892 stmtCarritoSel.close();

893 }

894 if (existe)

895 {

896 PreparedStatement stmtUpdateCar = conexion.prepareStatement(

897 sql: "UPDATE carrito_compra SET cantidad=? WHERE id_usuario=? AND id_articulo=?"

898);

899 try

900 {

901 stmtUpdateCar.setInt(parameterIndex: 1, cantidadEnCarrito + req.cantidad);

902 stmtUpdateCar.setInt(parameterIndex: 2, req.id_usuario);

903 stmtUpdateCar.setInt(parameterIndex: 3, req.id_articulo);

904 stmtUpdateCar.executeUpdate();

905 }

906 finally

907 {

908 stmtUpdateCar.close();

909 }

910 }

911 else

912 {

913 PreparedStatement stmtInsertCar = conexion.prepareStatement(

914 sql: "INSERT INTO carrito_compra(id_usuario,id_articulo,cantidad) VALUES (?, ?, ?)"

915);

916 try

917 {

918 stmtInsertCar.setInt(parameterIndex: 1, req.id_usuario);

919 stmtInsertCar.setInt(parameterIndex: 2, req.id_articulo);

920 stmtInsertCar.setInt(parameterIndex: 3, req.cantidad);

921 stmtInsertCar.executeUpdate();

922 }

923 finally

924 {

925 stmtInsertCar.close();

926 }

927 }

928 // Actualizar stock

Snipping Tool

Screenshot copied to clipboard

Automatically saved to screenshots folder.

Markup and share

Ln 725 Col 10 Spaces: 4 UFT-8 LF Java JS Put: 5000

02/29/2024 02:59 p.m.

ESP US

Wi-Fi

02/29/2024 02:59 p.m.

The screenshot shows an IDE interface with multiple tabs open. The active tab is 'Servicio.java' which contains Java code for a service class. The code includes methods for inserting items into a cart and updating stock levels. The IDE's status bar at the bottom right shows 'Java Ready', 'Port: 5500', and the date '02/12/2025'.

```
public class Servicio {
    public Response compraArticulo(CompraArticuloRequest req) throws Exception {
        try {
            statInsertCar.setInt(parameterIndex: 2, req.id_articulo);
            statInsertCar.setInt(parameterIndex: 3, req.cantidad);
            statInsertCar.executeUpdate();
        }
        finally {
            statInsertCar.close();
        }

        // Actualizar stock
        PreparedStatement statUpdateStock = conexion.prepareStatement(
            sql: "UPDATE stock SET cantidad = cantidad - ? WHERE id_articulo=?"
        );
        try {
            statUpdateStock.setInt(parameterIndex: 1, req.cantidad);
            statUpdateStock.setInt(parameterIndex: 2, req.id_articulo);
            statUpdateStock.executeUpdate();
        }
        finally {
            statUpdateStock.close();
        }

        conexion.commit();
        return Response.ok("Mensaje: \"Compra realizada\"").build();
    }
    catch (Exception e)
    {
        conexion.rollback();
        return Response.status(400).entity(j.writeValueAsString(new HuboError(e.getMessage()))).build();
    }
    finally
    {
        conexion.setAutoCommit(autoCommit: true);
        conexion.close();
    }
}

// 4) elimina_articulo_carrito_compra
@DELETE
@Path("elimina_articulo_carrito_compra")
```

4. Eliminar articulo de carrito de compra

The screenshot shows an IDE interface with multiple tabs open. The active tab is 'Servicio.java' which contains Java code for a service class. The code includes a method for deleting an item from the shopping cart. It checks if the user has access, then updates the database to reflect the change. The IDE's status bar at the bottom right shows 'Java Ready', 'Port: 5500', and the date '02/12/2025'.

```
// 4) elimina_articulo_carrito_compra
@DELETE
@Path("elimina_articulo_carrito_compra")
@Produces(MediaType.APPLICATION_JSON)
public Response eliminaArticuloCarritoCompras(
    @QueryParam("id_usuario") int id_usuario,
    @QueryParam("id_articulo") int id_articulo,
    @QueryParam("token") String token) throws Exception
{
    Connection conexion = pool.getConnection();

    if (!verifica_acceso(conexion, id_usuario, token))
    {
        conexion.close();
        return Response.status(400).entity(j.writeValueAsString(new HuboError(message: "Acceso denegado"))).build();
    }

    try
    {
        conexion.setAutoCommit(autoCommit: false);

        int cantidadCarrito = 0;

        PreparedStatement statSel = conexion.prepareStatement(
            sql: "SELECT cantidad FROM carrito_compra WHERE id_usuario=? AND id_articulo=? FOR UPDATE"
        );
        try
        {
            statSel.setInt(parameterIndex: 1, id_usuario);
            statSel.setInt(parameterIndex: 2, id_articulo);

            ResultSet rs = statSel.executeQuery();
            try
            {
                if (!rs.next())
                    throw new Exception(message: "El articulo no est&acute;a en el carrito");
                cantidadCarrito = rs.getInt(columnIndex: 1);
            }
            finally
            {
                rs.close();
            }
        }
        finally
        {
            statSel.close();
        }
    }
    finally
    {
        conexion.close();
    }
}
```

The screenshot shows an IDE interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Terminal:** Shows the command "tarea7".
- Code Editor:** Displays Java code for a service class named "Servicio". The method "public Response eliminaArticuloCarrito()" is shown, which performs an UPDATE operation on a database table "carrito_compra" to decrease the quantity of an item. It also performs a DELETE operation to remove the item from the user's shopping cart.
- Explorer:** Shows the project structure under "TAREA7". It includes a "Service" package with "Servicio.java", "META-INF", and "servicio" sub-folders containing various Java files like "AltaArticuloRequest.java", "Articulo.java", etc. It also includes "WEB-INF", "libs", and "OUTLINE" sections.
- Bottom Bar:** Shows system icons for battery, signal, and time (02:59 p.m., 02/12/2025).

5. Eliminar carrito de compra

The screenshot shows an IDE interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Terminal:** Shows the command "tarea7".
- Code Editor:** Displays Java code for a service class named "Servicio". The method "public Response eliminaArticuloCarrito()" is annotated with @DELETE and @Path("elimina_carrito_compra"). It takes parameters @QueryParam("id_usuario") int id_usuario and @QueryParam("token") String token. It checks if the user has access (verifica_acceso) and then closes the connection if they do not. It then tries to obtain the items in the cart using a prepared statement and adds them to a list.
- Explorer:** Shows the project structure under "TAREA7". It includes a "Service" package with "Servicio.java", "META-INF", and "servicio" sub-folders containing various Java files like "AltaArticuloRequest.java", "Articulo.java", etc. It also includes "WEB-INF", "libs", and "OUTLINE" sections.
- Bottom Bar:** Shows system icons for battery, signal, and time (02:59 p.m., 02/12/2025). A Snipping Tool window is open with the message "Screenshot copied to clipboard" and "Automatically saved to screenshots folder".

```

File Edit Selection View Go Run Terminal Help < -> tareas7
File Edit Selection View Go Run Terminal Help < -> tareas7
Articulo.java ItemCarrito.java AltaArticuloRequest.java CompraArticuloRequest.java Servicio.java <-> prueba.html ...&Servicio compruebaArticulo(CompraArticuloRequest)
META-INF <-> prueba.html ...&Servicio compruebaArticulo(CompraArticuloRequest)
service AltaArticuloRequest.class AltaArticuloRequest.java Articulo.java CompraArticuloRequest.class CompraArticuloRequest.java HuboError.class HuboError.java ItemCarrito.class ItemCarrito.java PingResource.java prueba.html prueba.html ...&Servicio colección.json
  META-INF context.xml
    service
      AltaArticuloRequest.class
      AltaArticuloRequest.java
      Articulo.java
      CompraArticuloRequest.class
      CompraArticuloRequest.java
      HuboError.class
      HuboError.java
      ItemCarrito.class
      ItemCarrito.java
      PingResource.java
      prueba.html
      prueba.html ...&Servicio
      prueba.html ...&Servicio
      Usario_en_foto.png
      Usario.java
      Usario.java
      WSClient.js
WEB-INF
  classes
  lib
    web.xml
  compila.bat
  compila.sh
  jackson-annotations-2.6.3.jar
  jackson-core-2.6.3.jar
  jackson-databind-2.6.3.jar
  jackson-jars-base-2.6.3.jar
  jackson-jars-json-provider-2.6.3.jar
  OUTLINE
  TIMELINE
  PROJECTS
  RUN CONFIGURATION
  JAVA PROJECTS
  286 56 Indexing completed.  Java Ready
  9+ 1058
  1059
  1060
  1061
  1062
  1063
  1064
  1065
  1066
  1067
  1068
  1069
  1070
  1071
  1072
  1073
  1074
  1075
  1076
  1077
  1078
  1079
  1080
  1081
  1082
  1083
  1084
  1085
  1086
  1087
  1088
  1089
  1090
  1091
  1092
  1093
  1094
  1095
  1096
  1097
  1098
  1099
  1100
  1101
  1102
  1103
  1104
  1105
  1106
  1107
  1108
  1109
  1110
  1111
  1112
  1113
  1114
  1115
  1116
  1117
  1118
  1119
  1120
  1121
  1122
  1123
  1124
  1125
  1126
  1127
  1128
  1129
  1130
  1131
  1132
  1133
  1134
  1135
  1136
  1137
  1138
  1139
  1140
  1141
  1142
  1143
  1144
  1145
  1146
  1147
  1148
  1149
  1150
  1151
  1152
  1153
  1154
  1155
  1156
  1157
  1158
  1159
  1160
  1161
  1162
  1163
  1164
  1165
  1166
  1167
  1168
  1169
  1170
  1171
  1172
  1173
  1174
  1175
  1176
  1177
  1178
  1179
  1180
  1181
  1182
  1183
  1184
  1185
  1186
  1187
  1188
  1189
  1190
  1191
  1192
  1193
  1194
  1195
  1196
  1197
  1198
  1199
  1200
  1201
  1202
  1203
  1204
  1205
  1206
  1207
  1208
  1209
  1210
  1211
  1212
  1213
  1214
  1215
  1216
  1217
  1218
  1219
  1220
  1221
  1222
  1223
  1224
  1225
  1226
  1227
  1228
  1229
  1230
  1231
  1232
  1233
  1234
  1235
  1236
  1237
  1238
  1239
  1240
  1241
  1242
  1243
  1244
  1245
  1246
  1247
  1248
  1249
  1250
  1251
  1252
  1253
  1254
  1255
  1256
  1257
  1258
  1259
  1260
  1261
  1262
  1263
  1264
  1265
  1266
  1267
  1268
  1269
  1270
  1271
  1272
  1273
  1274
  1275
  1276
  1277
  1278
  1279
  1280
  1281
  1282
  1283
  1284
  1285
  1286
  1287
  1288
  1289
  1290
  1291
  1292
  1293
  1294
  1295
  1296
  1297
  1298
  1299
  1300
  1301
  1302
  1303
  1304
  1305
  1306
  1307
  1308
  1309
  1310
  1311
  1312
  1313
  1314
  1315
  1316
  1317
  1318
  1319
  1320
  1321
  1322
  1323
  1324
  1325
  1326
  1327
  1328
  1329
  1330
  1331
  1332
  1333
  1334
  1335
  1336
  1337
  1338
  1339
  1340
  1341
  1342
  1343
  1344
  1345
  1346
  1347
  1348
  1349
  1350
  1351
  1352
  1353
  1354
  1355
  1356
  1357
  1358
  1359
  1360
  1361
  1362
  1363
  1364
  1365
  1366
  1367
  1368
  1369
  1370
  1371
  1372
  1373
  1374
  1375
  1376
  1377
  1378
  1379
  1380
  1381
  1382
  1383
  1384
  1385
  1386
  1387
  1388
  1389
  1390
  1391
  1392
  1393
  1394
  1395
  1396
  1397
  1398
  1399
  1400
  1401
  1402
  1403
  1404
  1405
  1406
  1407
  1408
  1409
  1410
  1411
  1412
  1413
  1414
  1415
  1416
  1417
  1418
  1419
  1420
  1421
  1422
  1423
  1424
  1425
  1426
  1427
  1428
  1429
  1430
  1431
  1432
  1433
  1434
  1435
  1436
  1437
  1438
  1439
  1440
  1441
  1442
  1443
  1444
  1445
  1446
  1447
  1448
  1449
  1450
  1451
  1452
  1453
  1454
  1455
  1456
  1457
  1458
  1459
  1460
  1461
  1462
  1463
  1464
  1465
  1466
  1467
  1468
  1469
  1470
  1471
  1472
  1473
  1474
  1475
  1476
  1477
  1478
  1479
  1480
  1481
  1482
  1483
  1484
  1485
  1486
  1487
  1488
  1489
  1490
  1491
  1492
  1493
  1494
  1495
  1496
  1497
  1498
  1499
  1500
  1501
  1502
  1503
  1504
  1505
  1506
  1507
  1508
  1509
  1510
  1511
  1512
  1513
  1514
  1515
  1516
  1517
  1518
  1519
  1520
  1521
  1522
  1523
  1524
  1525
  1526
  1527
  1528
  1529
  1530
  1531
  1532
  1533
  1534
  1535
  1536
  1537
  1538
  1539
  1540
  1541
  1542
  1543
  1544
  1545
  1546
  1547
  1548
  1549
  1550
  1551
  1552
  1553
  1554
  1555
  1556
  1557
  1558
  1559
  1560
  1561
  1562
  1563
  1564
  1565
  1566
  1567
  1568
  1569
  1570
  1571
  1572
  1573
  1574
  1575
  1576
  1577
  1578
  1579
  1580
  1581
  1582
  1583
  1584
  1585
  1586
  1587
  1588
  1589
  1590
  1591
  1592
  1593
  1594
  1595
  1596
  1597
  1598
  1599
  1600
  1601
  1602
  1603
  1604
  1605
  1606
  1607
  1608
  1609
  1610
  1611
  1612
  1613
  1614
  1615
  1616
  1617
  1618
  1619
  1620
  1621
  1622
  1623
  1624
  1625
  1626
  1627
  1628
  1629
  1630
  1631
  1632
  1633
  1634
  1635
  1636
  1637
  1638
  1639
  1640
  1641
  1642
  1643
  1644
  1645
  1646
  1647
  1648
  1649
  1650
  1651
  1652
  1653
  1654
  1655
  1656
  1657
  1658
  1659
  1660
  1661
  1662
  1663
  1664
  1665
  1666
  1667
  1668
  1669
  1670
  1671
  1672
  1673
  1674
  1675
  1676
  1677
  1678
  1679
  1680
  1681
  1682
  1683
  1684
  1685
  1686
  1687
  1688
  1689
  1690
  1691
  1692
  1693
  1694
  1695
  1696
  1697
  1698
  1699
  1700
  1701
  1702
  1703
  1704
  1705
  1706
  1707
  1708
  1709
  1710
  1711
  1712
  1713
  1714
  1715
  1716
  1717
  1718
  1719
  1720
  1721
  1722
  1723
  1724
  1725
  1726
  1727
  1728
  1729
  1730
  1731
  1732
  1733
  1734
  1735
  1736
  1737
  1738
  1739
  1740
  1741
  1742
  1743
  1744
  1745
  1746
  1747
  1748
  1749
  1750
  1751
  1752
  1753
  1754
  1755
  1756
  1757
  1758
  1759
  1760
  1761
  1762
  1763
  1764
  1765
  1766
  1767
  1768
  1769
  1770
  1771
  1772
  1773
  1774
  1775
  1776
  1777
  1778
  1779
  1780
  1781
  1782
  1783
  1784
  1785
  1786
  1787
  1788
  1789
  1790
  1791
  1792
  1793
  1794
  1795
  1796
  1797
  1798
  1799
  1800
  1801
  1802
  1803
  1804
  1805
  1806
  1807
  1808
  1809
  1810
  1811
  1812
  1813
  1814
  1815
  1816
  1817
  1818
  1819
  1820
  1821
  1822
  1823
  1824
  1825
  1826
  1827
  1828
  1829
  1830
  1831
  1832
  1833
  1834
  1835
  1836
  1837
  1838
  1839
  1840
  1841
  1842
  1843
  1844
  1845
  1846
  1847
  1848
  1849
  1850
  1851
  1852
  1853
  1854
  1855
  1856
  1857
  1858
  1859
  1860
  1861
  1862
  1863
  1864
  1865
  1866
  1867
  1868
  1869
  1870
  1871
  1872
  1873
  1874
  1875
  1876
  1877
  1878
  1879
  1880
  1881
  1882
  1883
  1884
  1885
  1886
  1887
  1888
  1889
  1890
  1891
  1892
  1893
  1894
  1895
  1896
  1897
  1898
  1899
  1900
  1901
  1902
  1903
  1904
  1905
  1906
  1907
  1908
  1909
  1910
  1911
  1912
  1913
  1914
  1915
  1916
  1917
  1918
  1919
  1920
  1921
  1922
  1923
  1924
  1925
  1926
  1927
  1928
  1929
  1930
  1931
  1932
  1933
  1934
  1935
  1936
  1937
  1938
  1939
  1940
  1941
  1942
  1943
  1944
  1945
  1946
  1947
  1948
  1949
  1950
  1951
  1952
  1953
  1954
  1955
  1956
  1957
  1958
  1959
  1960
  1961
  1962
  1963
  1964
  1965
  1966
  1967
  1968
  1969
  1970
  1971
  1972
  1973
  1974
  1975
  1976
  1977
  1978
  1979
  1980
  1981
  1982
  1983
  1984
  1985
  1986
  1987
  1988
  1989
  1990
  1991
  1992
  1993
  1994
  1995
  1996
  1997
  1998
  1999
  2000
  2001
  2002
  2003
  2004
  2005
  2006
  2007
  2008
  2009
  2010
  2011
  2012
  2013
  2014
  2015
  2016
  2017
  2018
  2019
  2020
  2021
  2022
  2023
  2024
  2025
  2026
  2027
  2028
  2029
  2030
  2031
  2032
  2033
  2034
  2035
  2036
  2037
  2038
  2039
  2040
  2041
  2042
  2043
  2044
  2045
  2046
  2047
  2048
  2049
  2050
  2051
  2052
  2053
  2054
  2055
  2056
  2057
  2058
  2059
  2060
  2061
  2062
  2063
  2064
  2065
  2066
  2067
  2068
  2069
  2070
  2071
  2072
  2073
  2074
  2075
  2076
  2077
  2078
  2079
  2080
  2081
  2082
  2083
  2084
  2085
  2086
  2087
  2088
  2089
  2090
  2091
  2092
  2093
  2094
  2095
  2096
  2097
  2098
  2099
  2100
  2101
  2102
  2103
  2104
  2105
  2106
  2107
  2108
  2109
  2110
  2111
  2112
  2113
  2114
  2115
  2116
  2117
  2118
  2119
  2120
  2121
  2122
  2123
  2124
  2125
  2126
  2127
  2128
  2129
  2130
  2131
  2132
  2133
  2134
  2135
  2136
  2137
  2138
  2139
  2140
  2141
  2142
  2143
  2144
  2145
  2146
  2147
  2148
  2149
  2150
  2151
  2152
  2153
  2154
  2155
  2156
  2157
  2158
  2159
  2160
  2161
  2162
  2163
  2164
  2165
  2166
  2167
  2168
  2169
  2170
  2171
  2172
  2173
  2174
  2175
  2176
  2177
  2178
  2179
  2180
  2181
  2182
  2183
  2184
  2185
  2186
  2187
  2188
  2189
  2190
  2191
  2192
  2193
  2194
  2195
  2196
  2197
  2198
  2199
  2200
  2201
  2202
  2203
  2204
  2205
  2206
  2207
  2208
  2209
  2210
  2211
  2212
  2213
  2214
  2215
  2216
  2217
  2218
  2219
  2220
  2221
  2222
  2223
  2224
  2225
  2226
  2227
  2228
  2229
  2230
  2231
  2232
  2233
  2234
  2235
  2236
  2237
  2238
  2239
  2240
  2241
  2242
  2243
  2244
  2245
  2246
  2247
  2248
  2249
  2250
  2251
  2252
  2253
  2254
  2255
  2256
  2257
  2258
  2259
  2260
  2261
  2262
  2263
  2264
  2265
  2266
  2267
  2268
  2269
  2270
  2271
  2272
  2273
  2274
  2275
  2276
  2277
  2278
  2279
  2280
  2281
  2282
  2283
  2284
  2285
  2286
  2287
  2288
  2289
  2290
  2291
  2292
  2293
  2294
  2295
  2296
  2297
  2298
  2299
  2300
  2301
  2302
  2303
  2304
  2305
  2306
  2307
  2308
  2309
  2310
  2311
  2312
  2313
  2314
  2315
  2316
  2317
  2318
  2319
  2320
  2321
  2322
  2323
  2324
  2325
  2326
  2327
  2328
  2329
  2330
  2331
  2332
  2333
  2334
  2335
  2336
  2337
  2338
  2339
  2340
  2341
  2342
  2343
  2344
  2345
  2346
  2347
  2348
  2349
  2350
  2351
  2352
  2353
  2354
  2355
  2356
  2357
  2358
  2359
  2360
  2361
  2362
  2363
  2364
  2365
  2366
  2367
  2368
  2369
  2370
  2371
  2372
  2373
  2374
  2375
  2376
  2377
  2378
  2379
  2380
  2381
  2382
  2383
  2384
  2385
  2386
  2387
  2388
  2389
  2390
  2391
  2392
  2393
  2394
  2395
  2396
  2397
  2398
  2399
  2400
  2401
  2402
  2403
  2404
  2405
  2406
  2407
  2408
  2409
  2410
  2411
  2412
  2413
  2414
  2415
  2416
  2417
  2418
  2419
  2420
  2421
  2422
  2423
  2424
  2425
  2426
  2427
  2428
  2429
  2430
  2431
  2432
  2433
  2434
  2435
  2436
  2437
  2438
  2439
  2440
  2441
  2442
  2443
  2444
  2445
  2446
  2447
  2448
  2449
  2450
  2451
  2452
  2453
  2454
  2455
  2456
  2457
  2458
  2459
  2460
  2461
  2462
  2463
  2464
  2465
  2466
  2467
  2468
  2469
  2470
  2471
  2472
  2473
  2474
  2475
  2476
  2477
  2478
  2479
  2480
  2481
  2482
  2483
  2484
  2485
  2486
  2487
  2488
  2489
  2490
  2491
  2492
  2493
  2494
  2495
  2496
  2497
  2498
  2499
  2500
  2501
  2502
  2503
  2504
  2505
  2506
  2507
  2508
  2509
  2510
  2511
  2512
  2513
  2514
  2515
  2516
  2517
  2518
  2519
  2520
  2521
  2522
  2523
  2524
  2525
  2526
  2527
  2528
  2529
  2530
  2531
  2532
  2533
  2534
  2535
  2536
  2537
  2538
  2539
  2540
  2541
  2542
  2543
  2544
  2545
  2546
  2547
  2548
  2549
  2550
  2551
  2552
  2553
  2554
  2555
  2556
  2557
  2558
  2559
  2560
  2561
  2562
  2563
  2564
  2565
  2566
  2567
  2568
  2569
  2570
  2571
  2572
  2573
  2574
  2575
  2576
  2577
  2578
  2579
  2580
  2581
  2582
  2583
  2584
  2585
  2586
  2587
  2588
  2589
  2590
  2591
  2592
  2593
  2594
  2595
  2596
  2597
  2598
  2599
  2600
  2601
  2602
  2603
  2604
  2605
  2606
  2607
  2608
  2609
  2610
  2611
  2612
  2613
  2614
  2615
  2616
  2617
  2
```

The screenshot shows an IDE interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar labeled "tarea7". The left sidebar displays the project structure under "TAREA7" and "SERVICIO". The main editor window contains Java code for a "Servicio" class, specifically the "modificaCarrito" method. The code uses JDBC to update the "carrito_compra" table and the "stock" table. It includes try-catch blocks, prepared statements, and result sets. The code is annotated with line numbers from 1098 to 1140. The status bar at the bottom right shows "Ln 725, Col 10" and "03:00 p.m. 02/12/2025".

```
1098     int cantidadCarrito = 0;
1099
1100     PreparedStatement stmtCarrito = conexion.prepareStatement(
1101         sql: "SELECT cantidad FROM carrito_compra WHERE id_usuario=? AND id_articulo=? FOR UPDATE"
1102     );
1103     try
1104     {
1105         stmtCarrito.setInt(parameterIndex: 1, id_usuario);
1106         stmtCarrito.setInt(parameterIndex: 2, id_articulo);
1107
1108         ResultSet rs = stmtCarrito.executeQuery();
1109         try
1110         {
1111             if (!rs.next())
1112                 throw new Exception(message: "El articulo no está en el carrito");
1113
1114         }
1115         finally
1116         {
1117             rs.close();
1118         }
1119     }
1120     finally
1121     {
1122         stmtCarrito.close();
1123     }
1124
1125     if (incremento == 1)
1126     {
1127         // Verificar stock
1128         int cantidadStock = 0;
1129         PreparedStatement stmtStock = conexion.prepareStatement(
1130             sql: "SELECT cantidad FROM stock WHERE id_articulo=? FOR UPDATE"
1131         );
1132         try
1133         {
1134             stmtStock.setInt(parameterIndex: 1, id_articulo);
1135             ResultSet rs = stmtStock.executeQuery();
1136             try
1137             {
1138                 if (!rs.next())
1139                     throw new Exception(message: "El articulo no existe");
1140
1141                 cantidadStock = rs.getInt(columnIndex: 1);
1142
1143             }
1144             finally
1145             {
1146                 rs.close();
1147             }
1148
1149         }
1150         finally
1151         {
1152             stmtStock.close();
1153         }
1154
1155         if (cantidadStock <= 0)
1156         {
1157             conexion.rollback();
1158             return Response.status(400).entity(j.writeValueAsString(new HuboError(message: "No hay suficientes articulos")));
1159         }
1160
1161         // Incrementar carrito
1162         PreparedStatement stmtUpdateCar = conexion.prepareStatement(
1163             sql: "UPDATE carrito_compra SET cantidad = cantidad + 1 WHERE id_usuario=? AND id_articulo=?"
1164         );
1165         try
1166         {
1167             stmtUpdateCar.setInt(parameterIndex: 1, id_usuario);
1168             stmtUpdateCar.setInt(parameterIndex: 2, id_articulo);
1169             stmtUpdateCar.executeUpdate();
1170         }
1171     }
1172 }
```

This screenshot is nearly identical to the one above, showing the same Java code for modifying a shopping cart. However, it includes a "Snipping Tool" overlay window in the bottom right corner. The overlay has a title bar "Snipping Tool" and a message area that says "Screenshot copied to clipboard" and "Automatically saved to screenshots folder". The status bar at the bottom right shows "Ln 725, Col 10" and "03:00 p.m. 02/12/2025".

```
1098     int cantidadCarrito = 0;
1099
1100     PreparedStatement stmtCarrito = conexion.prepareStatement(
1101         sql: "SELECT cantidad FROM carrito_compra WHERE id_usuario=? AND id_articulo=? FOR UPDATE"
1102     );
1103     try
1104     {
1105         stmtCarrito.setInt(parameterIndex: 1, id_usuario);
1106         stmtCarrito.setInt(parameterIndex: 2, id_articulo);
1107
1108         ResultSet rs = stmtCarrito.executeQuery();
1109         try
1110         {
1111             if (!rs.next())
1112                 throw new Exception(message: "El articulo no está en el carrito");
1113
1114         }
1115         finally
1116         {
1117             rs.close();
1118         }
1119     }
1120     finally
1121     {
1122         stmtCarrito.close();
1123     }
1124
1125     if (incremento == 1)
1126     {
1127         // Verificar stock
1128         int cantidadStock = 0;
1129         PreparedStatement stmtStock = conexion.prepareStatement(
1130             sql: "SELECT cantidad FROM stock WHERE id_articulo=? FOR UPDATE"
1131         );
1132         try
1133         {
1134             stmtStock.setInt(parameterIndex: 1, id_articulo);
1135             ResultSet rs = stmtStock.executeQuery();
1136             try
1137             {
1138                 if (!rs.next())
1139                     throw new Exception(message: "El articulo no existe");
1140
1141                 cantidadStock = rs.getInt(columnIndex: 1);
1142
1143             }
1144             finally
1145             {
1146                 rs.close();
1147             }
1148
1149         }
1150         finally
1151         {
1152             stmtStock.close();
1153         }
1154
1155         if (cantidadStock <= 0)
1156         {
1157             conexion.rollback();
1158             return Response.status(400).entity(j.writeValueAsString(new HuboError(message: "No hay suficientes articulos")));
1159         }
1160
1161         // Incrementar carrito
1162         PreparedStatement stmtUpdateCar = conexion.prepareStatement(
1163             sql: "UPDATE carrito_compra SET cantidad = cantidad + 1 WHERE id_usuario=? AND id_articulo=?"
1164         );
1165         try
1166         {
1167             stmtUpdateCar.setInt(parameterIndex: 1, id_usuario);
1168             stmtUpdateCar.setInt(parameterIndex: 2, id_articulo);
1169             stmtUpdateCar.executeUpdate();
1170         }
1171     }
1172 }
```

The screenshot shows an IDE interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** tareas7
- Code Editor:** Articulo.java (line 35 to 1190). The code implements a `modificaCarrito` method in the `Servicio` class. It handles stock management, including incrementing and decrementing item counts in the cart and database, and handling hubo errors.
- Explorer:** Shows the project structure with packages like `TAREA7`, `Servicio`, and `WEB-INF`, along with various Java files, XML files, and configuration files.
- Terminal:** Shows the command `java -jar TAREA7.jar`.
- Output:** Shows the output of the application running on port 5500.
- Snipping Tool:** A floating window showing a screenshot of the application's user interface.
- System Tray:** Shows icons for battery, signal, and date/time (03:00 p.m., 02/12/2025).

The screenshot shows an IDE interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** tareas7
- Code Editor:** Articulo.java (line 1073 to 1228). The code implements a `modificaCarrito` method in the `Servicio` class. It handles cart modification logic, including item addition and removal, and database operations like updating cart and deleting from compra.
- Explorer:** Shows the project structure with packages like `TAREA7`, `Servicio`, and `WEB-INF`, along with various Java files, XML files, and configuration files.
- Terminal:** Shows the command `java -jar TAREA7.jar`.
- Output:** Shows the output of the application running on port 5500.
- Snipping Tool:** A floating window showing a screenshot of the application's user interface.
- System Tray:** Shows icons for battery, signal, and date/time (03:00 p.m., 02/12/2025).

The screenshot shows an IDE interface with a dark theme. The left sidebar displays a project structure for 'TAREA7' containing 'META-INF', 'libs', and 'Servicio'. The 'Servicio' folder contains 'context.xml' and 'servicio' which includes 'AltaArticuloRequest.class', 'Articulo.class', 'CompraArticuloRequest.class', 'CompraArticuloRequest.java', 'HuboError.class', 'HuboError.java', 'ItemCarrito.class', 'ItemCarrito.java', 'PingResource.java', 'prueba.html', 'Servicio.class', 'Usuario.class', 'Usuario.java', and 'WSClient.js'. The central area shows a Java code editor for 'Servicio.java' with the following code:

```

135     public Response modificaCarrito()
136     {
137         try
138         {
139             statUpdateCar.close();
140         }
141         catch (SQLException e)
142         {
143             e.printStackTrace();
144         }
145         finally
146         {
147             statUpdateStock.close();
148         }
149         return Response.ok("El carrito se modificó");
150     }
151     catch (Exception e)
152     {
153         e.printStackTrace();
154         return Response.status(400).entity(j.writeValueAsString(new HuboError(e.getMessage()))).build();
155     }
156     finally
157     {
158         conexion.setAutoCommit(true);
159         conexion.close();
160     }
161     // T) consulta_carrito_compra (para el front)
162     @GET
163     @Path("consulta_carrito_compra")
164     @Produces(MediaType.APPLICATION_JSON)
165     public Response consultaCarrito(
166         @QueryParam("id_usuario") int id_usuario,
167         @QueryParam("token") String token) throws Exception

```

The status bar at the bottom indicates 'Indexing completed.' and 'Java: Ready'.

Ademas, para se crearon las siguientes clases para poder manejar las entidades de la base de datos:

The screenshot shows an IDE interface with a dark theme. The left sidebar displays a project structure for 'TAREA7' containing 'META-INF', 'libs', and 'Servicio'. The 'Servicio' folder contains 'context.xml' and 'servicio' which includes 'AltaArticuloRequest.class', 'AltaArticuloRequest.java', 'Articulo.class', 'CompraArticuloRequest.class', 'CompraArticuloRequest.java', 'HuboError.class', 'HuboError.java', 'ItemCarrito.class', 'ItemCarrito.java', 'PingResource.java', 'prueba.html', 'Servicio.class', 'Usuario.class', and 'Usuario.java'. The central area shows a Java code editor for 'AltaArticuloRequest.java' with the following code:

```

1 package servicio;
2
3 public class AltaArticuloRequest
4 {
5     public String nombre;
6     public String descripcion;
7     public Double precio;
8     public Integer cantidad;
9     public byte[] foto;
10    public Integer id_usuario;
11    public String token;
12 }
13
14

```

The status bar at the bottom indicates 'Indexing completed.' and 'Java: Ready'.

The screenshot shows a Java development environment with the following details:

- File Explorer:** Shows a project named "TAREA7" containing a "libs" folder and a "Servicio" package. The "Servicio" package contains a "META-INF" folder with "context.xml" and a "servicio" folder containing "Articulo.java", "AltaArticuloRequest.class", "AltaArticuloRequest.java", "Articulo.class", "CompraArticuloRequest.class", "CompraArticuloRequest.java", "HuboError.class", "HuboError.java", "ItemCarrito.class", "ItemCarrito.java", "PingResource.java", "prueba.html", "Servicio.class", "Servicio.java", "usuario_sin_foto.png", and "Usuario.class".
- Editor:** The "Articulo.java" file is open in the editor. The code defines a class "Articulo" with fields: id_articulo (Integer), nombre (String), descripcion (String), precio (Double), cantidad (Integer), and foto (byte[]). A tooltip for the "cantidad" field indicates it can represent existencias or cantidad en carrito.
- Bottom Bar:** Shows indexing completed, Java Ready, and system status including date and time (03:03 p.m. 02/12/2025).

The screenshot shows a Java development environment with the following details:

- File Explorer:** Shows a project named "TAREA7" containing a "libs" folder and a "Servicio" package. The "Servicio" package contains a "META-INF" folder with "context.xml" and a "servicio" folder containing "Articulo.java", "AltaArticuloRequest.class", "AltaArticuloRequest.java", "CompraArticuloRequest.class", "CompraArticuloRequest.java", "HuboError.class", "HuboError.java", "ItemCarrito.class", "ItemCarrito.java", "PingResource.java", "prueba.html", "Servicio.class", "Servicio.java", "usuario_sin_foto.png", and "Usuario.class".
- Editor:** The "ItemCarrito.java" file is open in the editor. The code defines a class "ItemCarrito" with fields: id_articulo (Integer), nombre (String), descripcion (String), precio (Double), cantidad (Integer), costo (Double), and foto (byte[]). A tooltip for the "costo" field indicates it is cantidad * precio.
- Bottom Bar:** Shows indexing completed, Java Ready, and system status including date and time (03:03 p.m. 02/12/2025). A "Snipping Tool" window is visible in the bottom right corner.

The screenshot shows an IDE interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** Search term: tarea7
- Toolbars:** Standard toolbar with icons for file operations.
- Left Sidebar:**
 - EXPLORER: Shows a project structure with a folder named "TAREAS" containing "libs", "Servicio", "META-INF", and "servicio". Inside "servicio", there are several Java files: AltaArticuloRequest.class, AltaArticuloRequest.java, Articulo.class, Articulo.java, CompraArticuloRequest.class, CompraArticuloRequest.java, HuboError.class, HuboError.java, ItemCarrito.class, ItemCarrito.java, PingResource.java, prueba.html, Servicio.class, Servicio.java, Usuario.class, and Usuario_sin_foto.png.
 - OUTLINE, TIMELINE, PROJECTS, RUN CONFIGURATION, JAVA PROJECTS.
- Central Area:** Code editor showing the following Java code for `CompraArticuloRequest`:

```
// Source code is decompiled from a .class file using FernFlower decompiler (from IntelliJ IDEA).
package servicio;

public class CompraArticuloRequest {
    public Integer id_articulo;
    public Integer cantidad;
    public Integer id_usuario;
    public String token;

    public CompraArticuloRequest() {
    }
}
```

- Bottom Status Bar:** Indexing completed, Java: Ready, Ln 1, Col 1, Spaces: 3, Port: 5500, 03:04 p.m., 02/12/2025.
- Taskbar:** Shows various application icons.

Compilacion

The screenshot shows a Windows PowerShell window with the following command history and output:

```
Administrator: Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instalar la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows
PS C:\WINDOWS\system32> $libs = "D:\Code\sistemas.distribuidos\tarea7\libs"
PS C:\WINDOWS\system32> $javawsfile = "https://repo1.maven.org/maven2/java/x.ws.rs-api/2.0.1/java.x.ws.rs-ap
PS C:\WINDOWS\system32> $javawsfile > Join-Path $libs "java.x.ws.rs-api-2.0.1.jar"
PS C:\WINDOWS\system32> Invoke-WebRequest -Uri $javawsfile -Outfile $javawsfile
PS C:\WINDOWS\system32> Get-ChildItem $libs | Where-Object Name -like "java.x.ws.rs"

Directorio: D:\Code\sistemas.distribuidos\tarea7\libs

Mode LastWriteTime Length Name
---- -----time----- ----
-a--- 01/12/2025 08:28 p. m. 115534 java.x.ws.rs-api-2.0.1.jar

PS C:\WINDOWS\system32> $tomcatlib = "C:\Apache-Tomcat-8.5.99\lib"
PS C:\WINDOWS\system32> Copy-Item $javawsfile $tomcatlib
PS C:\WINDOWS\system32> Get-ChildItem $tomcatlib | Where-Object Name -like "java.x.ws.rs"

Directorio: C:\Apache-Tomcat-8.5.99\lib

Mode LastWriteTime Length Name
---- -----time----- ----

```

```
F:\> cd "C:\Apache\Tomcat-8.5.99\lib"
PS C:\Apache\Tomcat-8.5.99\lib> $classPath = "C:\Apache\Tomcat-8.5.99\lib"
PS C:\Apache\Tomcat-8.5.99\lib> Copy-Item $classPath\*.* -Force
PS C:\Apache\Tomcat-8.5.99\lib> Get-ChildItem $classPath | Where-Object Name -like ".java*.ws.rs"
#
# Directorio: C:\apache-tomcat-8.5.99\lib
#
# Node          LastWriteTime           Length Name
#-- ...
#---- 01/12/2025 00:28 p. m.      115534 java.ws.rs-api-2.0.1.jar

PS C:\Apache\Tomcat-8.5.99\lib> Set-Location "D:\Code\sistemas_distribuidos\tareas\servicio\servicio"
PS C:\Code\sistemas_distribuidos\tareas\servicio\servicio>
PS D:\Code\sistemas_distribuidos\tareas\servicio\servicio> $classPath = "D:\Code\sistemas_distribuidos\tareas\servicio\servicio\classes\*"
PS D:\Code\sistemas_distribuidos\tareas\servicio\servicio> $libPath = "D:\Code\sistemas_distribuidos\tareas\servicio\servicio\libs\*"
PS D:\Code\sistemas_distribuidos\tareas\servicio\servicio> $javaPath = "D:\Code\sistemas_distribuidos\tareas\servicio\servicio\java\*"
PS D:\Code\sistemas_distribuidos\tareas\servicio\servicio> $jarPath = "D:\Code\sistemas_distribuidos\tareas\servicio\servicio\jars\*"

PS D:\Code\sistemas_distribuidos\tareas\servicio\servicio> New-Item -itemType Directory -Name "classesPath" -ItemType Null
PS D:\Code\sistemas_distribuidos\tareas\servicio\servicio> Get-ChildItem $classPath *.class -ErrorAction SilentlyContinue | Remove-Item -Force
PS D:\Code\sistemas_distribuidos\tareas\servicio\servicio> Get-ChildItem $libPath *.jar -ErrorAction SilentlyContinue | Remove-Item -Force
PS D:\Code\sistemas_distribuidos\tareas\servicio\servicio> # Usar TODOS los jars de libs que ahora ya tienen java.ws.rs
PS D:\Code\sistemas_distribuidos\tareas\servicio\servicio> $java = "D:\Code\sistemas_distribuidos\tareas\servicio\servicio\java"
PS D:\Code\sistemas_distribuidos\tareas\servicio\servicio> java -cp "$java\*;" servicio> java

PS D:\Code\sistemas_distribuidos\tareas\servicio\servicio>
```

Pruebas en postman del backend (local)

Crear usuario

The screenshot shows the Postman application interface. On the left sidebar, under 'My Workspace', there is a 'Collections' section containing various API collections like 'authentication api', 'Calificación API', 'CicloEscolar API', 'CORE API', 'distribuidos', 'Llaveremos', 'Llaveremos - API Gateway Collection', 'NTgroup', 'proyecto ingenieria', 'Pruebas funcionales de caja negra proy...', 'ReminderApp', 'Sancion', 'sanciones caja negra', 'Schedule API Collection', 'Schedule API Collection Copy', 'Servicios - Backend Sistemas Distribuidos', and 'Usuarios'. Below these are sections for 'Files' (BETA) and 'Cloud View'. The main workspace displays a collection named 'Servicio - Backend Sistemas Distribuidos / Usuarios / 1- Alta usuario (POST /alta_usuario)'. It contains a single POST request with the URL {{baseUrl}} /alta_usuario. The 'Body' tab shows a JSON response with the key 'mensaje' and the value 'Se dio de alta el usuario'. The bottom navigation bar includes icons for Postbot, Runner, Start Proxy, Cookies, Vault, and Trash.

Login

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various collections and environments. In the main workspace, a GET request is defined to the endpoint `((baseUri)) /login?email= {{email}} &password= {{password}}`. The 'Params' tab is selected, showing query parameters: 'email' with value '{{email}}' and 'password' with value '{{password}}'. Below the request, the response body is shown as JSON:

```
1 {  
2     "id_usuario": 1,  
3     "token": "7aC2wvUR6E0ntKuusa98"  
4 }
```

Ver usuario

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various collections and environments. In the main workspace, a GET request is defined to the endpoint `((baseUri)) /consulta_usuario?email= {{email}} &token= {{token}}`. The 'Params' tab is selected, showing query parameters: 'email' with value '{{email}}' and 'token' with value '{{token}}'. Below the request, the response body is shown as JSON:

```
1 {  
2     "email": "usuario1@test.com",  
3     "password": null,  
4     "nombre": "Pedro",  
5     "apellido_paterno": "Landeros",  
6     "apellido_materno": "Cortes",  
7     "fecha_nacimiento": "2000-01-14T18:00:00.000",  
8     "telefono": 5512345678,  
9     "genero": "M",  
10    "foto": null  
11 }
```

Modificar usuario

The screenshot shows the Postman interface with the 'My Workspace' collection selected. The 'PUT 4 - Modifica usuario (PUT /modifica_usuario)' request is highlighted. The 'Params' tab shows two parameters: 'email' with value '{{email}}' and 'token' with value '{{token}}'. The response body is JSON with the message: "mensaje": "Se modificó el usuario".

Alta articulo

The screenshot shows the Postman interface with the 'My Workspace' collection selected. The 'POST 6 - Alta articulo (POST /alta_articulo)' request is highlighted. The 'Params' tab shows no parameters. The response body is JSON with the message: "mensaje": "Se dio de alta el artículo".

Consulta artículos

My Workspace

Collections

Environments

History

Flows

Files (BETA)

Cloud View

Find and replace

Console

Search Postman

Ctrl + K

Invite

Upgrade

Service - Backend Sistemas Distribuidos / Artículos / 7 - Consulta artículos (GET /consulta_articulos)

Send

Docs Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
palabra	{{palabra}}		
id_usuario	{{id_usuario}}		
token	{{token}}		

Body Cookies Headers (5) Test Results

```
{
  "id_articulo": 1,
  "nombre": "Laptop Gamer",
  "descripcion": "Laptop potente para juegos",
  "precio": 15000.5,
  "cantidad": null,
  "foto": null
}
```

200 OK 68 ms 288 B Save Response

Postbot Runner Start Proxy Cookies Vault Trash ESP US 09:11 a.m. 02/12/2025

Comprar articulo

My Workspace

Collections

Environments

History

Flows

Files (BETA)

Cloud View

Find and replace

Console

Search Postman

Ctrl + K

Invite

Upgrade

Service - Backend Sistemas Distribuidos / Carrito y compras / 8 - Compra artículo (POST /compra_articulo)

Send

Docs Params Authorization Headers (10) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
-----	-------	-------------	-----------

Body Cookies Headers (5) Test Results

```
{
  "mensaje": "Compra realizada"
}
```

200 OK 44 ms 186 B Save Response

Postbot Runner Start Proxy Cookies Vault Trash ESP US 09:11 a.m. 02/12/2025

Consultar carrito

The screenshot shows the Postman interface with a collection named "Servicio - Backend Sistemas Distribuidos". Under the "Usuarios" folder, there is a GET request labeled "9 - Consulta carrito (GET /consulta_carrito_compra)". The "Params" tab is selected, showing query parameters: "id_usuario" with value "{{id_usuario}}", and "token" with value "{{token}}". The "Body" tab shows a JSON response with one item:

```
[{"id_articulo": 1, "nombre": "Laptop Gamer", "descripcion": "Laptop potente para juegos", "precio": 15000.5, "cantidad": 2, "costo": 30001.0, "foto": null}]
```

Modificar carrito

The screenshot shows the Postman interface with the same collection and folder structure. Under the "Artículos" folder, there is a PUT request labeled "10 - Modifica carrito (PUT /modifica_carrito_compra incremento=1)". The "Params" tab is selected, showing parameters: "id_usuario" with value "{{id_usuario}}", "id_articulo" with value "{{id_articulo}}", "incremento" with value "1", and "token" with value "{{token}}". The "Body" tab shows a JSON body with one item:

```
{"mensaje": "Se modificó el carrito"}
```

Eliminar articulo

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists collections, environments, flows, and files. The main workspace displays a DELETE request for removing an item from a shopping cart. The URL is `((baseUrl)) /elimina_articulo_carrito_compra?id_usuario={{id_usuario}} &id_articulo={{id_articulo}} &token={{token}}`. The 'Params' tab shows four parameters: id_usuario, id_articulo, and token, each with their respective values. The 'Body' tab contains a JSON response with the message: "mensaje": "Se eliminó el artículo del carrito". The status bar at the bottom indicates a 200 OK response with a duration of 27 ms and a size of 210 B.

Eliminar carrito

This screenshot is identical to the one above, showing the same DELETE request to remove an item from a shopping cart. The URL is `((baseUrl)) /elimina_articulo_carrito_compra?id_usuario={{id_usuario}} &id_articulo={{id_articulo}} &token={{token}}`. The 'Params' tab shows id_usuario, id_articulo, and token. The 'Body' tab shows the JSON response: "mensaje": "Se eliminó el artículo del carrito". The status bar at the bottom indicates a 200 OK response with a duration of 27 ms and a size of 210 B.

Configuracion de maquina virtual

Información general

Grupo de recursos (mover) : rg-pedro-2023630402
Estado : Detenido (desasignado)
Ubicación : East US 2 (Zona 1)
Suscripción (mover) : Azure for Students
Id. de suscripción : 81d3d13e-e29d-4bd2-be30-9103abfb7700
Zona de disponibilidad : 1

Propiedades

Máquina virtual
Nombre del equipo : T7-2023630402
Sistema operativo : Linux
Generación de VM : V2
Arquitectura de VM : x64
Hibernación : Deshabilitado

Redes
Dirección IP pública : 20.62.0.127 (Interfaz de red T7-2023630402623_x1)
Dirección IP pública (IPv6) : -
Dirección IP privada : 172.16.0.4
Dirección IP privada (IPv6) : -

Propiedades

Máquina virtual
Nombre del equipo : T7-2023630402
Sistema operativo : Linux
Generación de VM : V2
Arquitectura de VM : x64
Hibernación : Deshabilitado
Grupo host : -
Host : -
Grupo con ubicación por proximidad : -
Estado de ubicación : N/D
Grupo de reserva de capacidad : -
Tipo de controladora de disco : SCSI

Redes
Dirección IP pública : 20.62.0.127 (Interfaz de red T7-2023630402623_x1)
Dirección IP pública (IPv6) : -
Dirección IP privada : 172.16.0.4
Dirección IP privada (IPv6) : -
Red virtual/subred : vnet-eastus2/snet-eastus2-1
Nombre DNS : Configurar

Tamaño
Tamaño : Standard B1s
vCPU : 1
RAM : 1 GB

Detalles de la imagen de origen
Proveedor de la imagen de origen : canonical
Oferta de la imagen de origen : ubuntu-24.04-its
Plan de imagen de origen : server

T7-2023630402 - Microsoft Azure

Ayuda para copiar esta máquina virtual en cualquier región | Administrar esta VM con CLI de Azure

Información general | Registro de actividad | Control de acceso (IAM) | Etiquetas | Diagnóstico y solución de problemas | Visualizador de recursos | Conectar | Redes | Configuración de red | Equilibrio de carga | Grupos de seguridad de la aplicación | Administrador de red | Configuración | Disponibilidad y escala | Seguridad | Copia de seguridad y recuperación ante desastres | Operaciones | Supervisión

Grupo de reserva de capacidad - 1 | Tipo de controladora de disco - SCSI | Azure de acceso puntual - Azure de acceso puntual | Directiva de explosión de Azure de acceso puntual | Disponibilidad y escalado - Zona de disponibilidad (editar) 1 | Zona extendida | Conjunto de disponibilidad | Conjunto de escalado (asociar) | Seguridad - Tipo de seguridad: Inicio seguro | Habilitar arranque seguro: Habilitado | Habilitar vTPM: Habilitado | Supervisión de integridad: Deshabilitado | Details de la imagen de origen - Publisher: canonical | Offer: ubuntu-24.04-its | Plan de imagen de origen: server | Disco - Disk ID: T7-2023630402_OsDisk_1, Encrypted in host: Deshabilitado | Azure Disk Encryption: No habilitado | Disk of SO ephemeral: N/D | Discos de datos: 0 | Apagado automático - Apagado automático: No habilitado | Apagado programado: | Detalles de la configuración de red - Servicio: Custom | Intervalos de puertos de destino: 8080 | Protocolo: Any | Firewall rules: Rule 300 (SSH) - Port: 22, Protocol: TCP, Action: Permitir | Rule 320 (HTTP) - Port: 80, Protocol: TCP, Action: Permitir | Rule 65000 (AllowVnetInbound) - Port: 65000, Protocol: Any, Action: Permitir | Rule 65001 (AllowAzureLoadBalancerInbound) - Port: 65001, Protocol: Any, Action: Permitir | Rule 65500 (DenyAllInbound) - Port: 65500, Protocol: Any, Action: Denegar

Reglas de entrada

Regla puerto http 8080

Agregar regla de seguridad de entrada

Dirección IP pública : 20.62.0.127 | Dirección IP privada : 172.16.0.4 | Reglas de seguridad de administrador : 0 (Configurar)

Reglas

Grupo de seguridad de r | Redes aceleradas | Reglas de seguridad vigente

Protocolo: Any | TCP | UDP | ICMPv4 | ICMPv6

Nombre: allow-http-8080 | Port: 8080 | Action: Permitir | Denegar

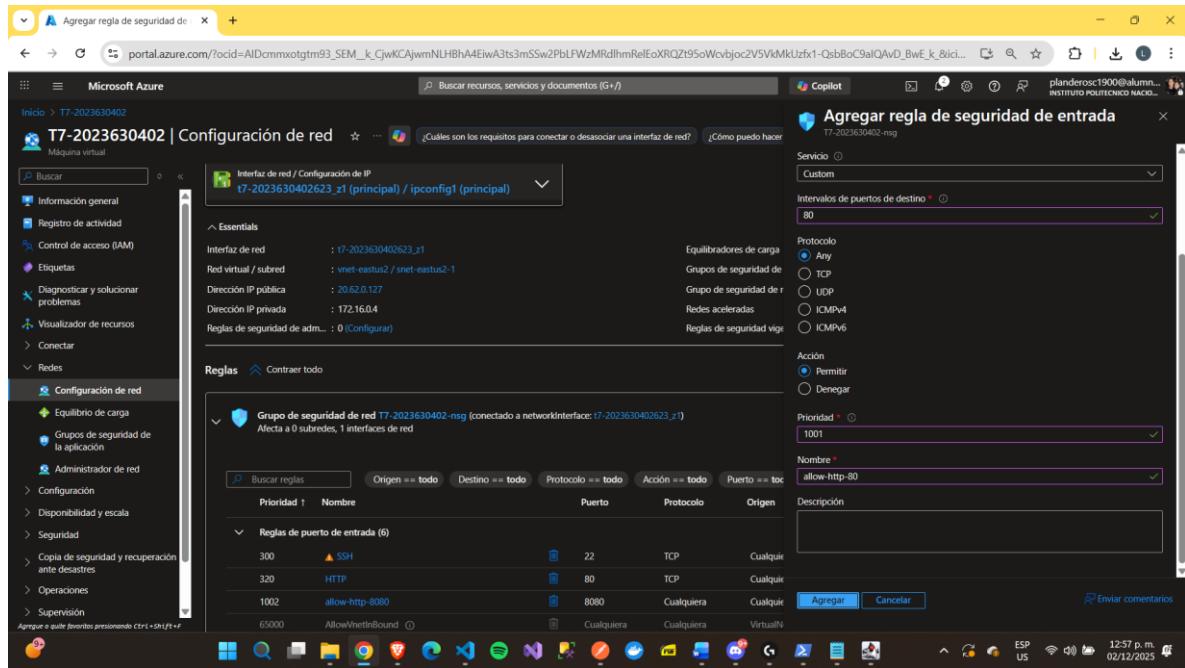
Origen: | Puerto: | Protocolo: | Acción: | Prioridad: 1002 | Descripción:

Reglas de puerto de entrada (5)

300 SSH | 320 HTTP | 65000 AllowVnetInbound | 65001 AllowAzureLoadBalancerInbound | 65500 DenyAllInbound

Reglas de puerto de salida (3)

Regla puerto http 80



Configuracion de la base de datos

Tablas antes

reporte.docx

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> USE servicio_web;
Reading table information for completion of table and column names.
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_servicio_web |
+-----+
| fotos_usuarios |
| usuarios |
+-----+
2 rows in set (0.00 sec)

mysql>
```

Tablas antes:

Comandos

Agregar campos a usuarios:

```
ALTER TABLE usuarios  
    ADD COLUMN password VARCHAR(64) NOT NULL AFTER email,  
    ADD COLUMN token  VARCHAR(20) NULL AFTER password;
```

Crear table stock:

```
CREATE TABLE stock (  
    id_articulo INT AUTO_INCREMENT PRIMARY KEY,  
    nombre      VARCHAR(100) NOT NULL,  
    descripcion VARCHAR(255),  
    precio      DECIMAL(10,2) NOT NULL,  
    cantidad    INT NOT NULL  
);
```

Crear tabla fotos:

```
CREATE TABLE fotos_articulos (  
    id_foto    INT AUTO_INCREMENT PRIMARY KEY,  
    foto       LONGBLOB,  
    id_articulo INT NOT NULL,  
    CONSTRAINT fk_fotos_articulos_stock  
        FOREIGN KEY (id_articulo) REFERENCES stock(id_articulo)  
        ON DELETE CASCADE  
);
```

Crear tabla carrito:

```
CREATE TABLE carrito_compra (  
    id_usuario  INT NOT NULL,
```

```

    id_articulo INT NOT NULL,
    cantidad  INT NOT NULL,
    PRIMARY KEY (id_usuario, id_articulo),
    CONSTRAINT fk_carrito_usuario
        FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)
        ON DELETE CASCADE,
    CONSTRAINT fk_carrito_articulo
        FOREIGN KEY (id_articulo) REFERENCES stock(id_articulo)
        ON DELETE CASCADE
);

```

Levantar servicio

```

# 1. Copiar proyecto desde Windows a la VM
# (Ejecutar en PowerShell local)
# =====
scp -i "C:\Users\lande\Downloads\T7-2023630402_key.pem" -r "D:\Code\sistemas
distribuidos\tarea7" azureuser@20.62.0.127:~

# =====
# 2. Conectarse a la máquina virtual por SSH
# =====
ssh -i "C:\Users\lande\Downloads\T7-2023630402_key.pem" azureuser@20.62.0.127

```

```
# =====
# 3. Entrar al directorio del proyecto
# =====
cd ~/tarea7/Servicio/Servicio

# 5. Configurar Tomcat
# =====
export CATALINA_HOME=/home/azureuser/apache-tomcat-8.5.99

# Copiar librerías Jackson/Jersey
cp jackson*.jar jersey-entity-filtering-2.24.jar jersey-media-json-jackson-2.24.jar
$CATALINA_HOME/lib/

# =====
# 6. Editar context.xml (MANUAL)
# =====
# nano META-INF/context.xml
# username="hugo"
# password="TU_PASSWORD"
# luego Ctrl+O, Enter, Ctrl+X

# =====
# 7. Editar compila.sh (MANUAL)
# =====
# nano compila.sh
# export CATALINA_HOME=/home/azureuser/apache-tomcat-8.5.99
```

```
# luego Ctrl+O, Enter, Ctrl+X
chmod +x compila.sh

# =====
# 8. Compilar servicio (genera Servicio.war)
# =====
sh compila.sh

# =====
# 9. Apagar Tomcat antes de desplegar
# =====
sh $CATALINA_HOME/bin/catalina.sh stop
pkill -f tomcat

# =====
# 10. Borrar WAR anterior de Tomcat
# =====
rm -rf $CATALINA_HOME/webapps/Servicio
rm -f $CATALINA_HOME/webapps/Servicio.war

# =====
# 11. Copiar el nuevo Servicio.war
# =====
cp Servicio.war $CATALINA_HOME/webapps/

# =====
```

```
# 12. Copiar front a ROOT  
# =====  
cp prueba.html WSClient.js usuario_sin_foto.png $CATALINA_HOME/webapps/ROOT/  
  
# =====  
# 13. Levantar Tomcat  
# =====  
sh $CATALINA_HOME/bin/catalina.sh start  
  
# =====  
# 14. Revisar logs para confirmar que no haya errores  
# =====  
tail -n 50 $CATALINA_HOME/logs/catalina.out  
  
# =====  
# 15. Probar localmente dentro de la VM  
# =====  
curl http://localhost:8080/prueba.html
```

Probar endpoints

POST 1 - Alta usuario (POST /alta_usuario)

```
POST /((baseUri))/alta_usuario
http://20.62.0.127:8080/Servicio/rest/ws/((baseUri))/alta_usuario
```

Body

```
{
  "email": "{{email}}",
  "password": "{{password}}",
  "nombre": "Pedro",
  "apellido_paterno": "Landeros",
  "apellido_materno": "Cortes",
  "fecha_nacimiento": "2000-01-15T00:00:00.000",
  "telefono": "+521917MKA7R"
}
```

200 OK

Body

```
{"mensaje": "Se dio de alta el usuario"}
```

GET 2 - Login (GET /login)

```
GET /((baseUri))/login?email={{email}}&password={{password}}
```

Query Params

Key	Value	Description
email	[[email]]	
password	[[password]]	

Body

```
{
  "id_usuario": 4,
  "token": "UreeSJQE4Quo0Ef0lWco"
}
```

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, Environments, History, Flows, Files (BETA).
- Header:** Home, Workspaces, API Network, Search Postman, Invite, Upgrade.
- Request:** POST 1 - Alta usuario (POST /alta_usuario), GET 2 - Login (GET /login), GET 3 - Consulta usuario (GET /consulta_usuario), PUT 4 - Modifica usuario (PUT /modifica_usuario), POST 6 - Alta artículo (POST /alta_articulo), GET 7 - Consulta artículos (GET /consulta_articulos), POST 8 - Compra artículo (POST /compra_articulo), GET 9 - Consulta carro (GET /consulta_carro), PUT 10 - Modifica carro (PUT /modifica_carro), PUT 11 - Modifica carro (PUT /modifica_carro), DEL 12 - Elimina artículo del carro (DELETE /borra_articulo), DEL 13 - Elimina carro completo (DELETE /borra_carro), shoes.
- Current Request:** GET 3 - Consulta usuario (GET /consulta_usuario) with URL {{baseUrl}} /consulta_usuario?email= {{(email)}} &token= {{(token)}}.
- Params:** email: {{(email)}}, token: {{(token)}}.
- Body:** JSON response (200 OK):

```
1 {
2   "email": "usuario1@test.com",
3   "password": null,
4   "nombre": "Pedro",
5   "apellido_paterno": "Landeros",
6   "apellido_materno": "Cortes",
7   "fecha_nacimiento": "2000-01-15T00:00:00.000",
8   "telefono": 5512345678,
9   "genero": "M",
10  "foto": null
11 }
```

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, Environments, History, Flows, Files (BETA).
- Header:** Home, Workspaces, API Network, Search Postman, Invite, Upgrade.
- Request:** POST 1 - Alta usuario (POST /alta_usuario), GET 2 - Login (GET /login), GET 3 - Consulta usuario (GET /consulta_usuario), PUT 4 - Modifica usuario (PUT /modifica_usuario), POST 6 - Alta artículo (POST /alta_articulo), GET 7 - Consulta artículos (GET /consulta_articulos), POST 8 - Compra artículo (POST /compra_articulo), GET 9 - Consulta carro (GET /consulta_carro), PUT 10 - Modifica carro (PUT /modifica_carro), PUT 11 - Modifica carro (PUT /modifica_carro), DEL 12 - Elimina artículo del carro (DELETE /borra_articulo), DEL 13 - Elimina carro completo (DELETE /borra_carro), shoes.
- Current Request:** PUT 4 - Modifica usuario (PUT /modifica_usuario) with URL {{baseUrl}} /modifica_usuario?email= {{(email)}} &token= {{(token)}}.
- Headers:** raw (selected), Content-Type: application/json.
- Body:** JSON response (200 OK):

```
1 {
2   "mensaje": "Se modificó el usuario"
3 }
```

My Workspace

POST 1 - Alta usuario (POST /alta_usuario)

```
1 "id_usuario": {{id_usuario}},
2 "token": "{{token}}",
3 "nombre": "Laptop Gamer 2",
4 "descripcion": "Laptop potente para juegos",
5 "precio": 15990.50,
6 "cantidad": 10
```

Body Cookies Headers (5) Test Results

200 OK 152 ms 197 B Save Response

POST 6 - Alta artículo (POST /alta_articulo)

```
1 {
2   "mensaje": "Se dio de alta el artículo"
```

Body Cookies Headers (5) Test Results

200 OK 152 ms 197 B Save Response

POST 1 - Alta usuario (POST /alta_usuario)

GET 2 - Login (GET /login)

GET 3 - Consulta usuario (GET /co...

PUT 4 - Modifica usuario (PUT /mo...

DEL 5 - Borra usuario (DELETE /bo...

POST 6 - Alta artículo (POST /alta_a...

GET 7 - Consulta artículos (GET /c...

POST 8 - Compra artículo (POST /co...

GET 9 - Consulta carrito (GET /con...

PUT 10 - Modifica carrito (PUT /mo...

PUT 11 - Modifica carrito (PUT /mo...

DEL 12 - Elimina artículo del carri...

DEL 13 - Elimina carrito completo (...

shoes

Cloud View Find and replace Console

Postbot Runner Start Proxy Cookies Vault Trash

ESP US 02:05 p.m. 02/12/2025

My Workspace

GET 7 - Consulta artículos (GET /consulta_articulos)

Params

Key	Value	Description
palabra		
id_usuario	{{id_usuario}}	
token	{{token}}	

Body Cookies Headers (5) Test Results

200 OK 11

```
1 [
2   {
3     "id_articulo": 1,
4     "nombre": "Test 1",
5     "descripcion": "test",
6     "precio": 10.0,
7     "cantidad": null,
8     "foto": null
9   },
10  {
11    "id_articulo": 2,
12    "nombre": "test 2"
```

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Markup and share

POST 1 - Alta usuario (POST /alta_usuario)

GET 2 - Login (GET /login)

GET 3 - Consulta usuario (GET /co...

PUT 4 - Modifica usuario (PUT /mo...

DEL 5 - Borra usuario (DELETE /bo...

POST 6 - Alta artículo (POST /alta_a...

GET 7 - Consulta artículos (GET /c...

POST 8 - Compra artículo (POST /co...

GET 9 - Consulta carrito (GET /con...

PUT 10 - Modifica carrito (PUT /mo...

PUT 11 - Modifica carrito (PUT /mo...

DEL 12 - Elimina artículo del carri...

DEL 13 - Elimina carrito completo (...

shoes

Cloud View Find and replace Console

Postbot Runner Start Proxy Cookies Vault Trash

ESP US 02:05 p.m. 02/12/2025

The screenshot shows the Postman interface with a collection named "Servicio - Backend Sistemas Distribuidos". The "Artículos" folder contains a POST request to `({baseUri})/compra_articulo`. The request body is set to "raw" JSON:

```
1 {
2   "id_usuario": "{{id_usuario}}",
3   "token": "{{token}}",
4   "id_articulo": "{{id_articulo}}",
5   "cantidad": 2
6 }
```

The response status is 200 OK with a response time of 97 ms and a size of 186 B. The response body is:

```
1 [
2   {
3     "mensaje": "Compra realizada"
4   }
5 ]
```

The screenshot shows the Postman interface with the same collection and "Artículos" folder. A GET request to `({baseUri})/consulta_carrito_compra?id_usuario={{id_usuario}}&token={{token}}` is selected. The "Params" tab shows query parameters:

Key	Value	Description
<input checked="" type="checkbox"/> id_usuario	<code>{{id_usuario}}</code>	
<input checked="" type="checkbox"/> token	<code>{{token}}</code>	

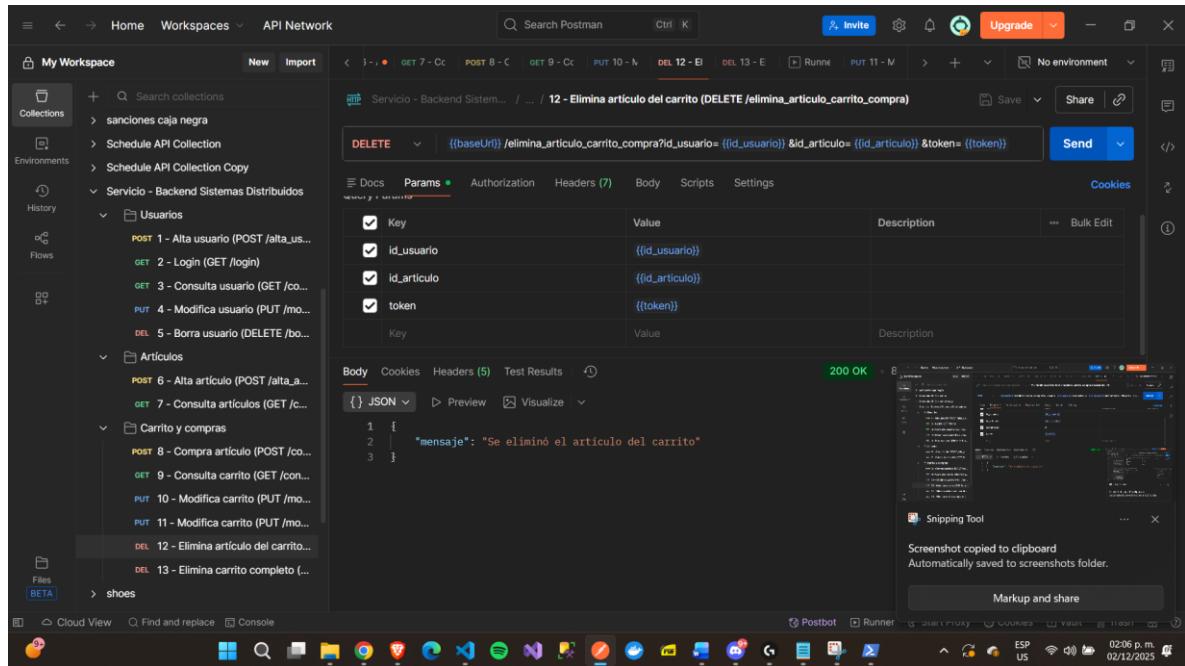
The "Body" tab shows the response body:

```
1 [
2   {
3     "id_articulo": 1,
4     "nombre": "Test 1",
5     "descripcion": "test",
6     "precio": 10.0,
7     "cantidad": 7,
8     "coste": 70.0,
9     "foto": null
10   }
11 ]
```

A message at the bottom indicates "Screenshot copied to clipboard" and "Automatically saved to screenshots folder".

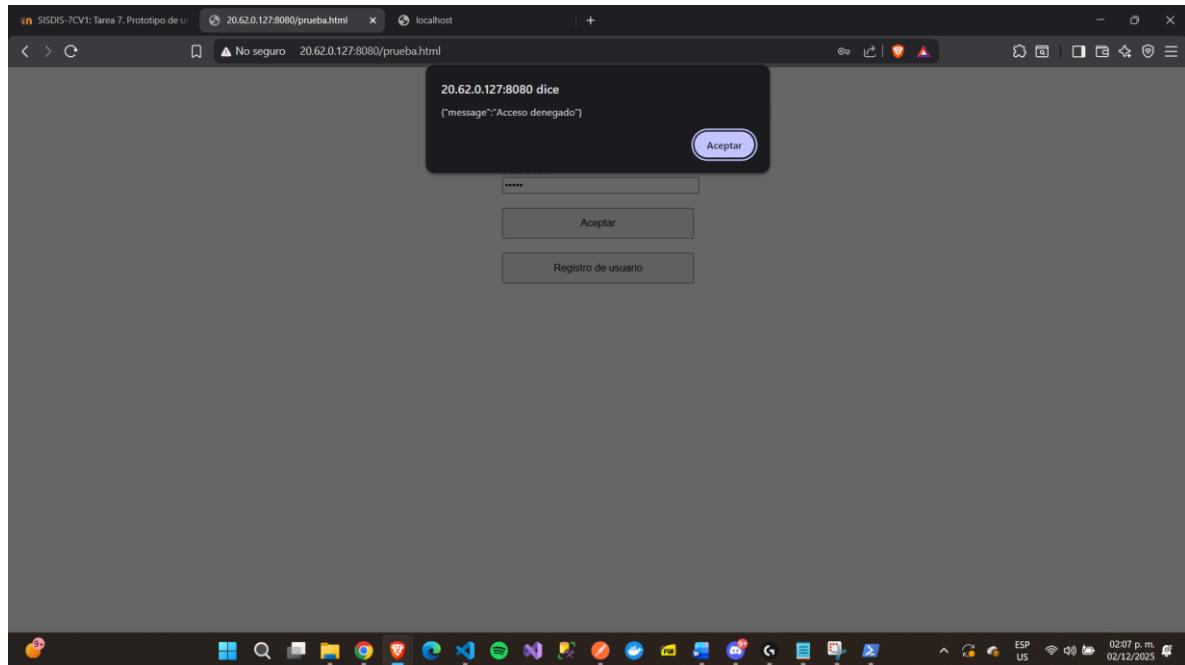
The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists collections, environments, flows, and files. The main workspace displays a collection named 'Servicio - Backend Sistemas Distribuidos'. Under this collection, there are two main sections: 'Usuarios' and 'Artículos'. The 'Artículos' section contains several requests, including 'POST 6 - Alta artículo (POST /alta_a...)', 'GET 7 - Consulta artículos (GET /co...', and 'PUT 10 - Modifica artículo (PUT /mo...'. A specific PUT request is selected, which has the URL `((baseUri)) /modifica_carrito_compra?id_usuario={{id_usuario}}&id_articulo={{id_articulo}}&incremento=1&token={{token}}`. The 'Params' tab shows parameters: id_usuario, id_articulo, incremento (with value 1), and token. The 'Body' tab shows a JSON response with the key "mensaje": "Se modificó el carrito". The status bar at the bottom indicates a 200 OK response.

This screenshot is nearly identical to the one above, showing the same Postman interface and workspace. The difference is in the selected PUT request's body parameters. In the 'Params' tab, the 'incremento' parameter is now set to -1. The rest of the interface, including the selected request, the JSON response, and the status bar, remains the same.



Pruebas del front

Login





Perfil

Perfil del usuario

Email *
a@gmail.com

Contraseña *
[redacted]

Nombre *
a

Apellido paterno *
b

Apellido materno
c

Fecha de nacimiento *
04/12/2025 07:39 p. m.

Teléfono
11223344

Género
Masculino

Elegir archivos Sin archivos seleccionados
Quitar foto

Guardar cambios

Regresar

Snipping Tool window:
Screenshot copied to clipboard
Automatically saved to screenshots folder.
Markup and share



4:47



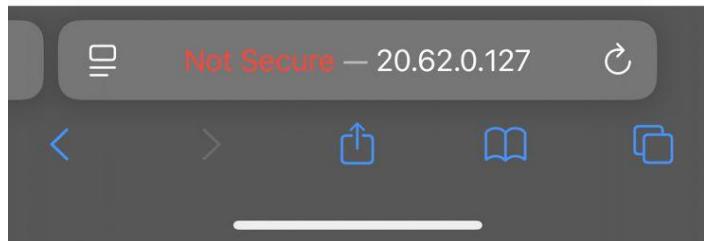
Perfil del usuario

Captura de artículo

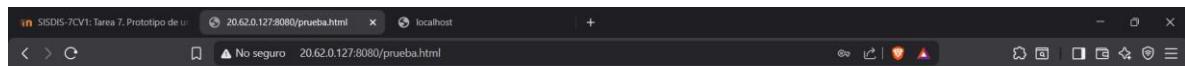
Compra de artículos

Eliminar perfil

Salir



1. Captura de articulo



Captura de artículo

Nombre *

Descripción

Precio *

Cantidad en existencia *

1

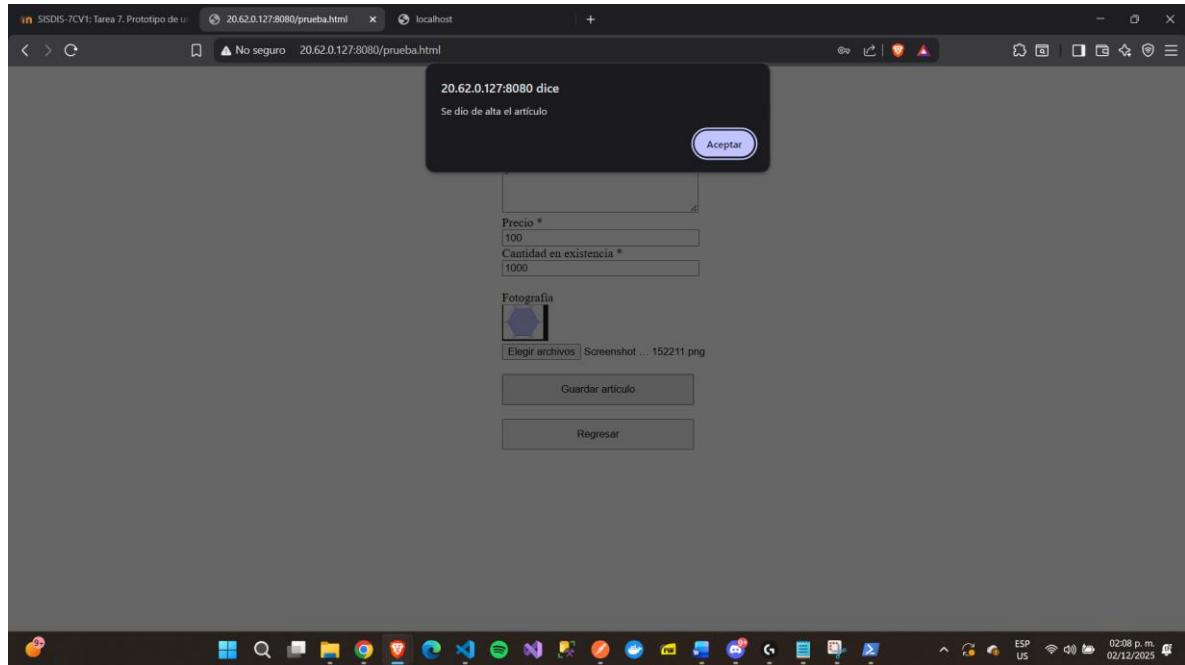
Fotografía



Elegir archivos Sin archivos seleccionados

Guardar artículo

Regresar



4:47



Captura de artículo

Nombre *

Test celular

Descripción

Precio *

100

Cantidad en existencia *

1000

Fotografía



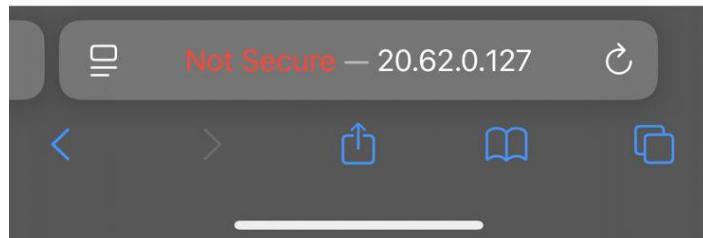
Choose Files



IMG_6314.jpeg

Guardar artículo

Regresar



4:47



Captura de artículo

Nombre *

Descripción

Precio *

Cantidad en existencia *

Se dio de alta el artículo

[Close](#)

 Choose Files  IMG_6314.jpeg

[Guardar artículo](#)

[Regresar](#)

Not Secure — 20.62.0.127

< >   

2. Comprar artículos y 3. Mostrar datos de artículos

SISDIS-7CV1: Tarea 7. Prototipo de un sistema de compras

20.62.0.127:8080/prueba.html

localhost

No seguro 20.62.0.127:8080/prueba.html

Compra de artículos

Palabra clave

Buscar

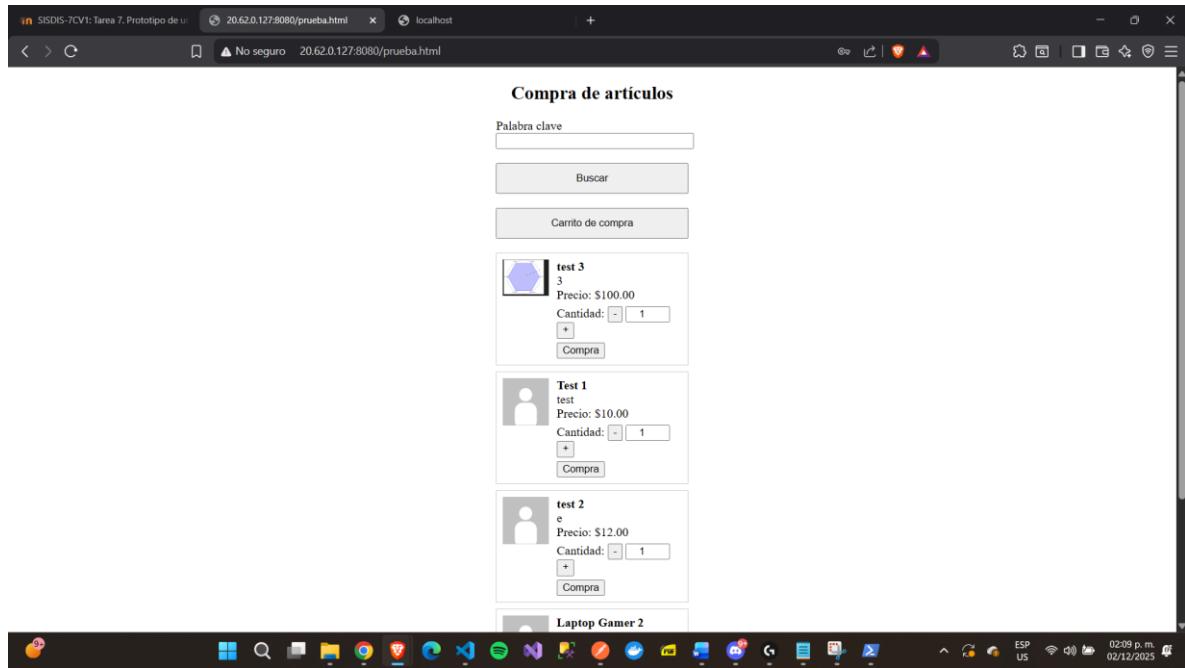
Carrito de compra

test 3
3
Precio: \$100.00
Cantidad: 1
 Compra

Test 1
test
Precio: \$10.00
Cantidad: 1
 Compra

test 2
e
Precio: \$12.00
Cantidad: 1
 Compra

Laptop Gamer 2



02:09 p. m. 02/12/2025

4:48



Compra de artículos

Palabra clave

Buscar

Carrito de compra

test 3
3
Precio: \$100.00
Cantidad: - 1 +

Compra

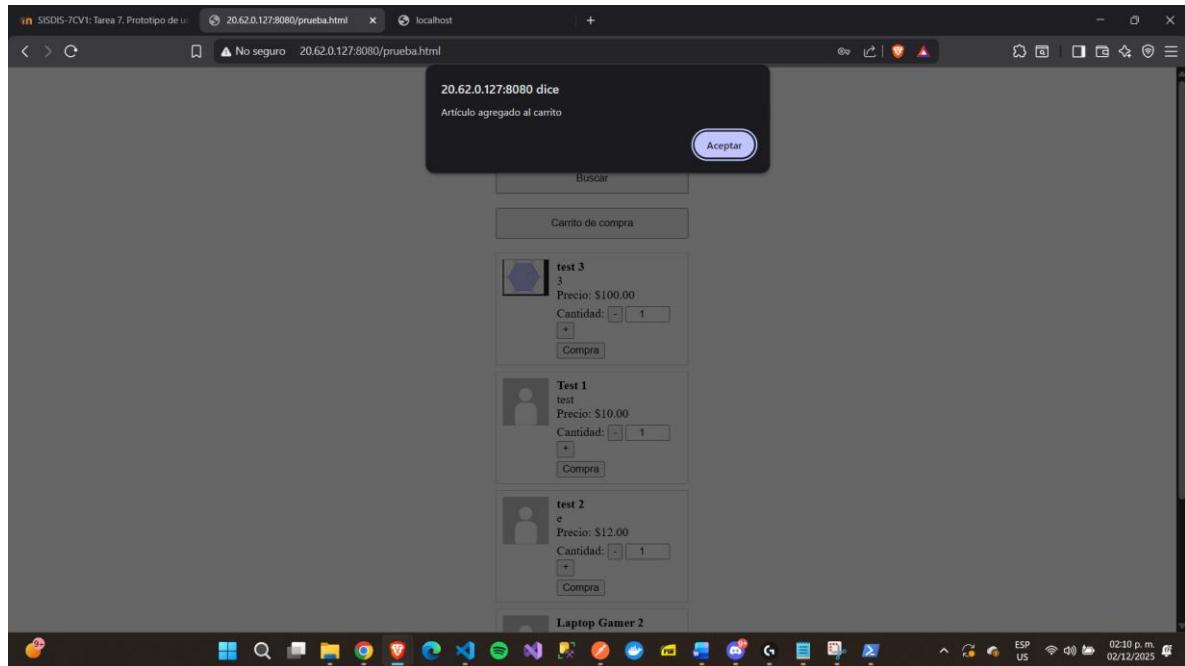
Test 1
test
Precio: \$10.00
Cantidad: - 1 +

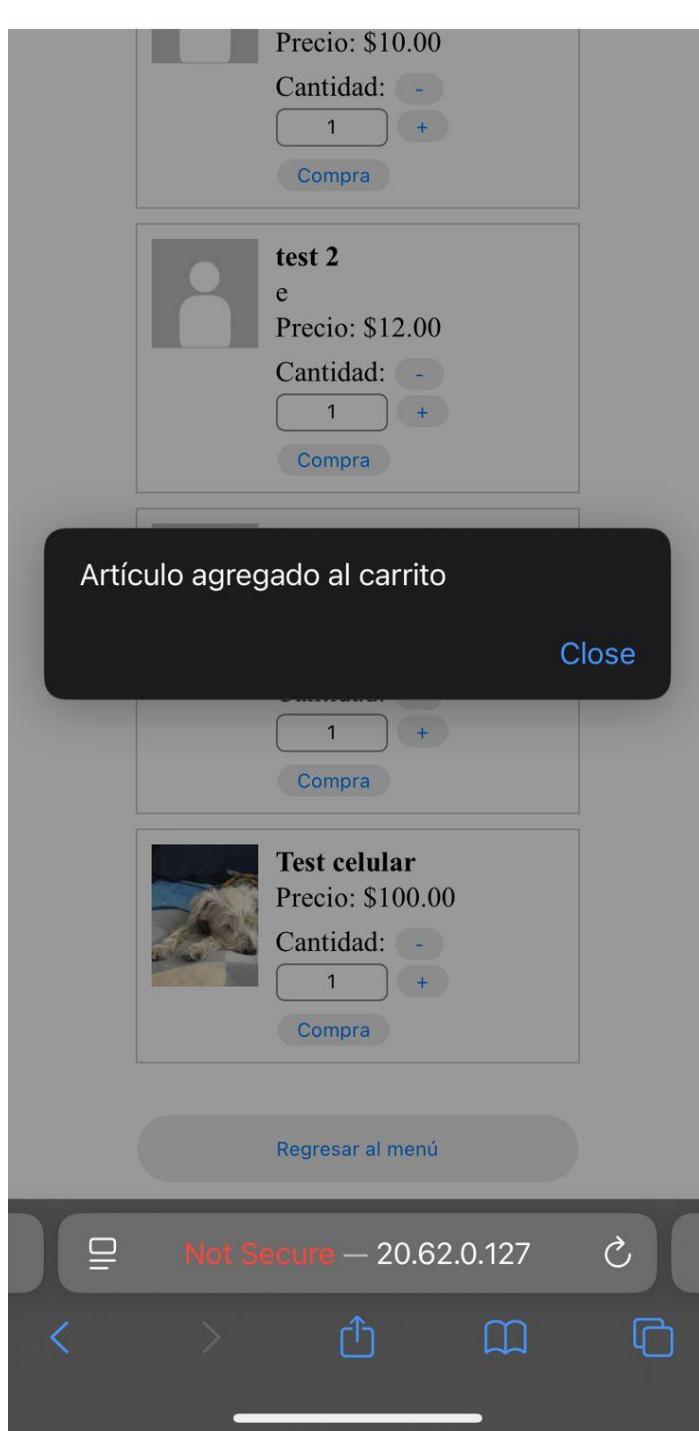
Compra

test 2
e
Precio: \$12.00
Cantidad: -



4. Presionar botón comprar





5, 6, y 7: Carrito de compras



4:48



Artículos en el carrito



Test celular
Precio: \$100.00
Cantidad: 1
Costo: \$100.00

+ -

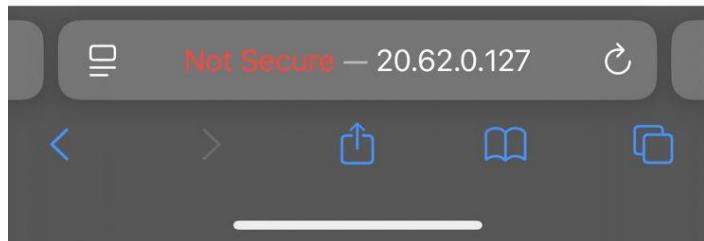
[Eliminar artículo](#)

Total: \$100.00

[Eliminar carrito](#)

[Seguir comprando](#)

[Regresar al menú](#)



4:48



Artículos en el carrito

No hay artículos en el carrito.

Total: \$0.00

[Eliminar carrito](#)

[Seguir comprando](#)

[Regresar al menú](#)



8. Seguir comprando

SISDIS-7CV1: Tarea 7. Prototipo de un sistema de compras

20.62.0.127:8080/prueba.html

localhost

No seguro 20.62.0.127:8080/prueba.html

Compra de artículos

Palabra clave

Buscar

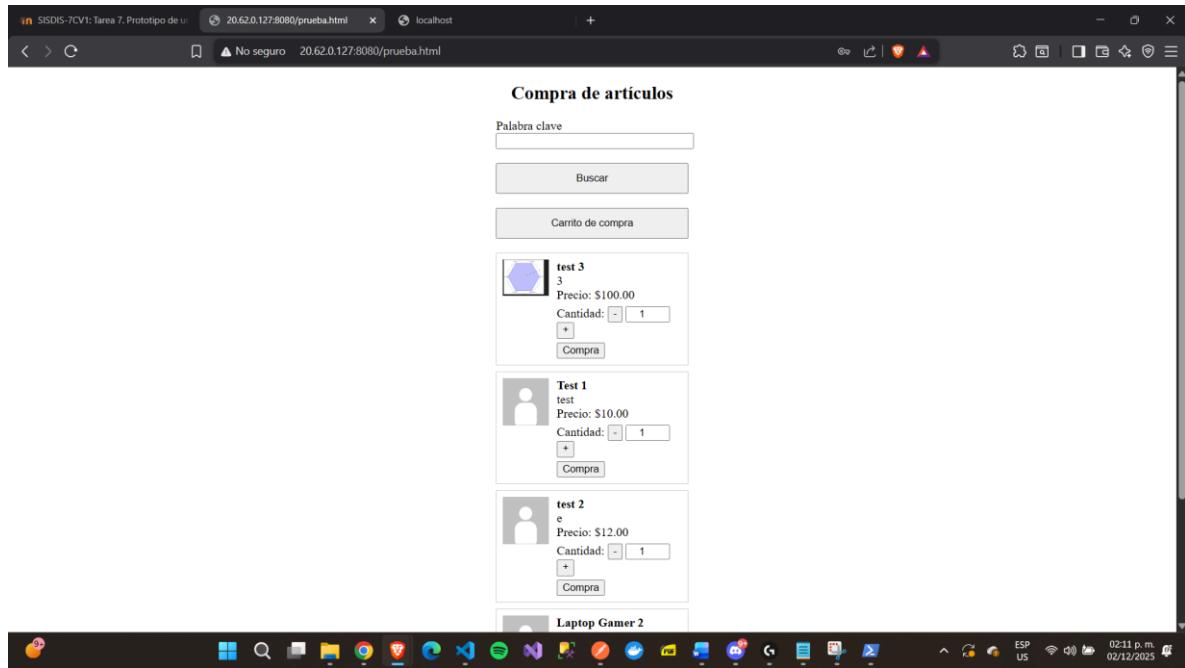
Carrito de compra

test 3
3
Precio: \$100.00
Cantidad: 1
 Compra

Test 1
test
Precio: \$10.00
Cantidad: 1
 Compra

test 2
e
Precio: \$12.00
Cantidad: 1
 Compra

Laptop Gamer 2



02:11 p. m. 02/12/2025

4:48



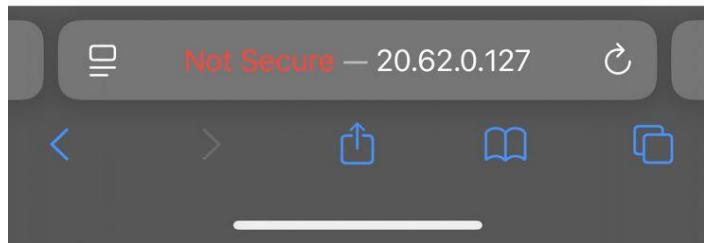
Compra de artículos

Palabra clave

Buscar

Carrito de compra

Regresar al menú



Conclusiones

El desarrollo de esta práctica me permitió comprender mejor cómo funciona realmente un sistema web cuando se despliega en un entorno similar al de producción. A través del proceso, pude darme cuenta de la importancia de cada paso: desde la creación correcta de una máquina virtual, hasta la configuración del servidor, las bases de datos y el despliegue del servicio web. Todo esto me hizo entender que no basta con que el código “funcione en mi máquina”, sino que debe poder ejecutarse de manera estable en un servidor real.

También pude entender de una forma más clara cómo interactúan las distintas capas del sistema. Al crear las tablas necesarias en MySQL para manejar artículos, fotos y carritos de compra, confirmé la importancia del diseño correcto de la base de datos y del uso de transacciones para mantener la integridad de la información. Asimismo, al compilar el servicio, generar el archivo WAR y subirlo al servidor Tomcat, experimenté cómo se hace un despliegue completo y los problemas que pueden surgir si algo queda mal configurado, como errores de permisos, rutas equivocadas o puertos ocupados.

Otro aprendizaje importante fue la interacción entre el front-end y el back-end mediante peticiones REST. Ver cómo la página prueba.html se comunicaba con los métodos del servicio y cómo estos respondían, me permitió comprender mejor el flujo completo de una aplicación web. Además, enfrentar errores reales, como procesos que quedaban corriendo, conflictos de puertos o fallas en Tomcat. Me ayudó a tener más criterio para diagnosticar y resolver problemas, en lugar de solo seguir pasos sin entenderlos.

En general, esta práctica me ayudó a reforzar conocimientos que antes tenía solo de forma teórica, y me permitió vivir el proceso completo de construir, configurar, desplegar y probar un servicio web funcional. Fue una experiencia que me hizo sentir más seguro trabajando con servidores, bases de datos, despliegue de aplicaciones y la integración de todos estos componentes en un sistema coherente.