

*Instituto Politécnico Nacional*  
Escuela Superior de Cómputo

Desarrollo de sistemas distribuidos

**Reporte Tarea 8.**  
**"Desarrollo de un prototipo de sistema de comercio electrónico utilizando un servicio web estilo REST"**

Grupo: 4CV13

**Integrantes:**

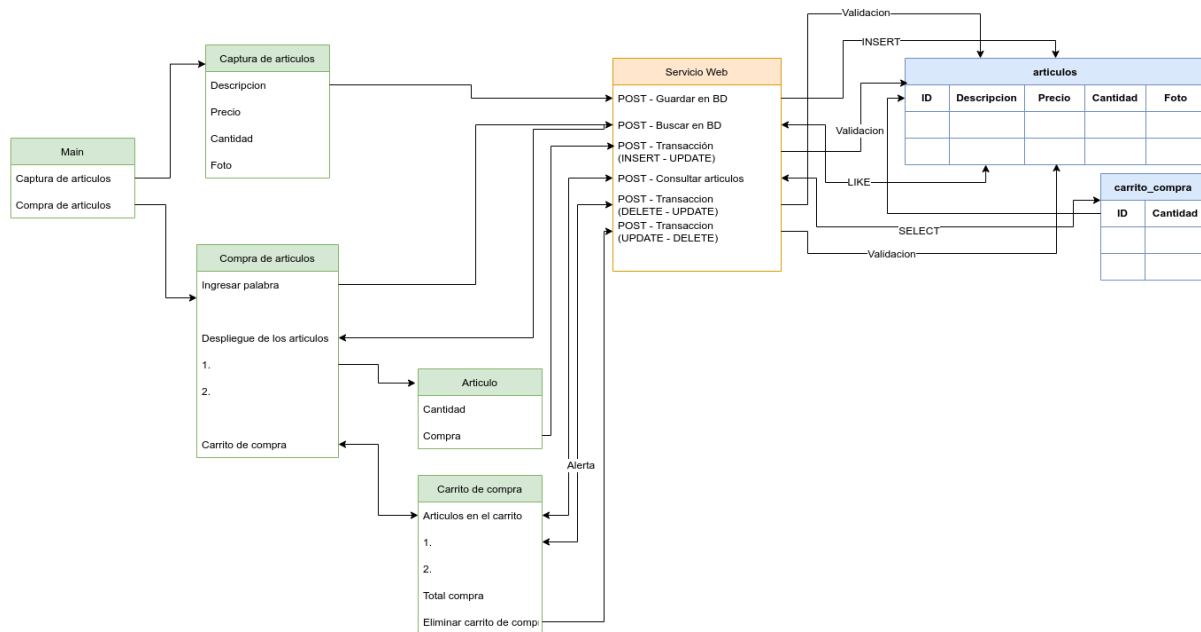
**Elisa Ramos Gomez**

Mayo, 2022.

# Desarrollo.

Para esta tarea se tuvo que desarrollar un cliente y un servicio web que es lo que hemos estado haciendo durante estas últimas clases, donde la idea es realizar el front-end y back-end, desde cero, donde tenemos que cubrir los 10 requerimientos funcionales.

Como apenas son mis primeros acercamientos en el tema, para comprender la tarea tuve que comprender lo que se tenía que hacer a través de la realización de un diagrama donde se pudiera apreciar de manera clara lo que se tenía que hacer, al final el diagrama quedó así:



## 1. Servicio.

Esta función de nombre `alta_articulo`, recibe como parámetro un objeto artículo serializado como Gson, haciendo a su vez un insert en la base de datos, donde si el insert fue exitoso regresa un código de OK por el contrario regresa un código de error; esto se ve de la siguiente manera:

```
@POST
@Path("alta_articulo")
@Consumes(MediaType.APPLICATION_FORM_URLENCODED)
@Produces(MediaType.APPLICATION_JSON)
public Response alta2(@FormParam("articulo") Artículo articulo) throws
Exception {

    Connection conexion = pool.getConnection();

    if (articulo.descripcion == null || articulo.descripcion.equals(""))
        return Response.status(400).entity(j.toJson(new Error("Se debe
    ingresar la descripcion"))).build();

    if (articulo.precio == null || articulo.precio.equals(""))
        return Response.status(400).entity(j.toJson(new Error("Se debe
```

```

    ingresar el precio"))).build();

    if (articulo.cantidad < 0)
        return Response.status(400).entity(j.toJson(new Error("Se debe
    ingresar una cantidad correcta"))).build();

    try {

        PreparedStatement stmt_1 = conexion.prepareStatement("INSERT INTO
    articulos VALUES (0,?, ?, ?, ?)");
        try {
            stmt_1.setString(1, articulo.descripcion);
            stmt_1.setString(2, articulo.precio);
            stmt_1.setInt(3, articulo.cantidad);
            stmt_1.setBytes(4, articulo.foto);

            stmt_1.executeUpdate();

        } finally {

            stmt_1.close();

        }

    } catch (Exception e) {

        return Response.status(400).entity(j.toJson(new
    Error(e.getMessage()))).build();

    } finally {

        conexion.close();

    }

    return Response.ok().build();

}

```

La siguiente función se llama `consulta_descripcion` donde recibe una string la cual es la palabra “buscar” en el campo de la table de artículos, para la consulta se usa la clausula like de SQL y regresa un Json con los artículos encontrados si es que existen de lo contrario lo regresa vacío e igual se maneja un código de error que se manda a impresión:

```

@POST
@Path("consulta_descripcion")
@Consumes(MediaType.APPLICATION_FORM_URLENCODED)
@Produces(MediaType.APPLICATION_JSON)
public Response consulta2(@FormParam("descripcion") String descripcion)
throws Exception {

    Connection conexion = pool.getConnection();
    if (descripcion == null || descripcion.equals(""))
        return Response.status(400).entity(j.toJson(new Error("Se debe
    ingresar la descripcion"))).build();

    try {

        //PreparedStatement stmt_1 = conexion.prepareStatement("SELECT * FROM
    articulos WHERE descripcion LIKE ?");
        String SQLstatement = "SELECT * FROM articulos WHERE descripcion

```

```

LIKE \' + descripcion + " %\';
    //PreparedStatement stmt_1 = conexion.prepareStatement("SELECT * FROM
    articulos WHERE descripcion LIKE '%%'");
    PreparedStatement stmt_1 = conexion.prepareStatement(SQLstatement);

    try {

        //stmt_1.setString(1, descripcion);

        ResultSet rs = stmt_1.executeQuery();

        try {

            List <Articulo> listaArticulos = new ArrayList<Articulo>();

            while (rs.next()) {

                Articulo articulo = new Articulo();
                articulo.descripcion = rs.getString(2);
                articulo.precio = rs.getString(3);
                articulo.cantidad = rs.getInt(4);
                articulo.foto = rs.getBytes(5);

                listaArticulos.add(articulo);

                //int n = rs.getFetchSize();
                //return Response.ok().entity(j.toJson(n)).build();

                /*
                Articulo [] articulos = new Articulo[n];

                for(int i = 0; i < n; i++){
                    articulos[i].descripcion = rs.getString(1+5*i);
                    articulos[i].precio = rs.getString(2+5*i);
                    articulos[i].cantidad = rs.getInt(3+5*i);
                    articulos[i].foto = rs.getBytes(4+5*i);
                }

                //Usuario r = new Usuario();

                //r.email = rs.getString(1);
                //r.nombre = rs.getString(2);
                //r.apellido_paterno = rs.getString(3);
                //r.apellido_materno = rs.getString(4);
                //r.fecha_nacimiento = rs.getString(5);
                //r.telefono = rs.getString(6);
                //r.genero = rs.getString(7);
                //r.foto = rs.getBytes(8);

                return Response.ok().entity(j.toJson(articulos)).build();
                */
            }

            return Response.ok().entity(j.toJson(listaArticulos)).build();
            //return Response.status(400).entity(j.toJson(new Error("El email
            no existe"))).build();

        } finally {
            rs.close();
        }

    } finally {
        stmt_1.close();
    }

```

```

    }

    } catch (Exception e) {

        return Response.status(400).entity(j.toJson(new
Error(e.getMessage()))).build();

    } finally {

        conexion.close();

    }

}
}

```

## 2. Front- End.

Para está parte re-utilizamos el WSCClient que se tenía desde la Tarea 6 donde la magia está en el HTML donde se agregaron etiquetas con sus respectivas descripciones al igual el manejo del DOM de forma dinámica para el despliegue de artículos,

## Creación de la máquina virtual.

Lo primero que se hace ingresando las especificaciones de la máquina virtual, como se muestra a continuación, claramente partiendo de la imagen creada en las tareas pasadas, cuando terminamos toda la configuración tenemos que dar click en revisar y crear.

Actividades Brave Web Browser 6 de may 20:57

Problema - omegaUp ReporteTarea8 - Documento: Crear una máquina virtual

portal.azure.com/#create/Microsoft.VirtualMachine-ARM

Inicio - Micros... mecanografía Feedback Helper Distribuidos

Microsoft Azure Buscar recursos, servicios y documentos (G+/)

eramosg1400@alumno... INSTITUTO POLITÉCNICO NACIONAL

Inicio > Distribuidos > Crear un recurso >

### Crear una máquina virtual

Datos básicos Discos Redes Administración Opciones avanzadas Etiquetas Revisar y crear

Cree una máquina virtual que ejecute Linux o Windows. Seleccione una imagen de Azure Marketplace o use una imagen personalizada propia. Complete la pestaña Conceptos básicos y, después, use Revisar y crear para aprovisionar una máquina virtual con parámetros predeterminados o bien revise cada una de las pestañas para personalizar la configuración. [Más información](#)

**Detalles del proyecto**

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción \* Azure for Students

Grupo de recursos \* Distribuidos [Crear nuevo](#)

**Detalles de instancia**

Nombre de máquina virtual \* CE2015021066

Región (US) East US

Opciones de disponibilidad No se requiere redundancia de la infraestructura

Tipo de seguridad Estándar

Imagen \* R2015021066-image-20220411 - Gen2 [Ver todas las imágenes](#) [Configurar la generación de máquinas virtuales](#)

Instancia de Azure de acceso puntual

Tamaño \* Standard\_B1s - 1 vcpu, 1 GiB de memoria (USD 7.59/mes) [Ver todos los tamaños](#)

**Cuenta de administrador**

Tipo de autenticación Clave pública SSH

[Revisar y crear](#) < Anterior Siguiente: Discos >

Actividades

Brave Web Browser

6 de may 20:58

Problema - omegaUp

ReporteTarea8 - Documento...

Crear una máquina virtual

portal.azure.com/#create/Microsoft.VirtualMachine-ARM

Inicio - Micros... mecanografía Feedback Helper Distribuidos

Microsoft Azure

Buscar recursos, servicios y documentos (G+/)

eramosg1400@alumno... INSTITUTO POLITÉCNICO NACIONAL

Inicio > Distribuidos > Crear un recurso >

Crear una máquina virtual

Tipo de seguridad

Imagen

Instancia de Azure de acceso puntual

Tamaño

Cuenta de administrador

Tipo de autenticación

Nombre de usuario

Contraseña

Confirmar contraseña

Reglas de puerto de entrada

Puertos de entrada públicos

Seleccionar puertos de entrada

Estándar

R2015021066-image-20220411 - Gen2

☐

Standard\_B1s - 1 vcpu, 1 GiB de memoria (USD 7.59/mes)

☐ Clave pública SSH

☒ Contraseña

ElisaRG

\*\*\*\*\*

\*\*\*\*\*

Seleccione los puertos de red de máquina virtual que son accesibles desde la red Internet pública. Puede especificar acceso de red más limitado o granular en la pestaña Red.

☐ Ninguno

☒ Permitir los puertos seleccionados

SSH (22)

Esto permitirá que todas las direcciones IP accedan a la máquina virtual. Esto solo se recomienda para las pruebas. Use los controles avanzados de la pestaña Redes a fin de crear reglas para limitar el tráfico entrante a las direcciones IP conocidas.

Revisar y crear

< Anterior

Siguiente: Discos >

Actividades

Brave Web Browser

6 de may 20:59

Problema - omegaUp

ReporteTarea8 - Documento...

Crear una máquina virtual

portal.azure.com/#create/Microsoft.VirtualMachine-ARM

Inicio - Micros... mecanografía Feedback Helper Distribuidos

Microsoft Azure

Buscar recursos, servicios y documentos (G+/)

eramosg1400@alumno... INSTITUTO POLITÉCNICO NACIONAL

Inicio > Distribuidos > Crear un recurso >

Crear una máquina virtual

Datos básicos

Discos

Redes

Administración

Opciones avanzadas

Etiquetas

Revisar y crear

Las máquinas virtuales de Azure tienen un disco de sistema operativo y un disco temporal para el almacenamiento a corto plazo. Puede asociar discos de datos adicionales. El tamaño de la máquina virtual determina el tipo de almacenamiento que puede usar y la cantidad de datos que permiten los discos. [Mas información](#)

Opciones de disco

Tipo de disco del sistema operativo

Eliminar con VM

Cifrado en el host

Tipo de cifrado

Habilitar compatibilidad con Ultra Disks

Discos de datos para CE2015021066

HDD estándar (almacenamiento con redundancia local)

☐

☐

(Predeterminado) Cifrado en reposo con una clave administrada por la plata...

☐

Puede agregar y configurar discos de datos adicionales para su máquina virtual o asociar discos existentes. Esta máquina virtual también incluye un disco temporal.

LUN	Nombre	Tamán...	Tipo de disco	Almacena...	Eliminar con VM
-----	--------	----------	---------------	-------------	-----------------

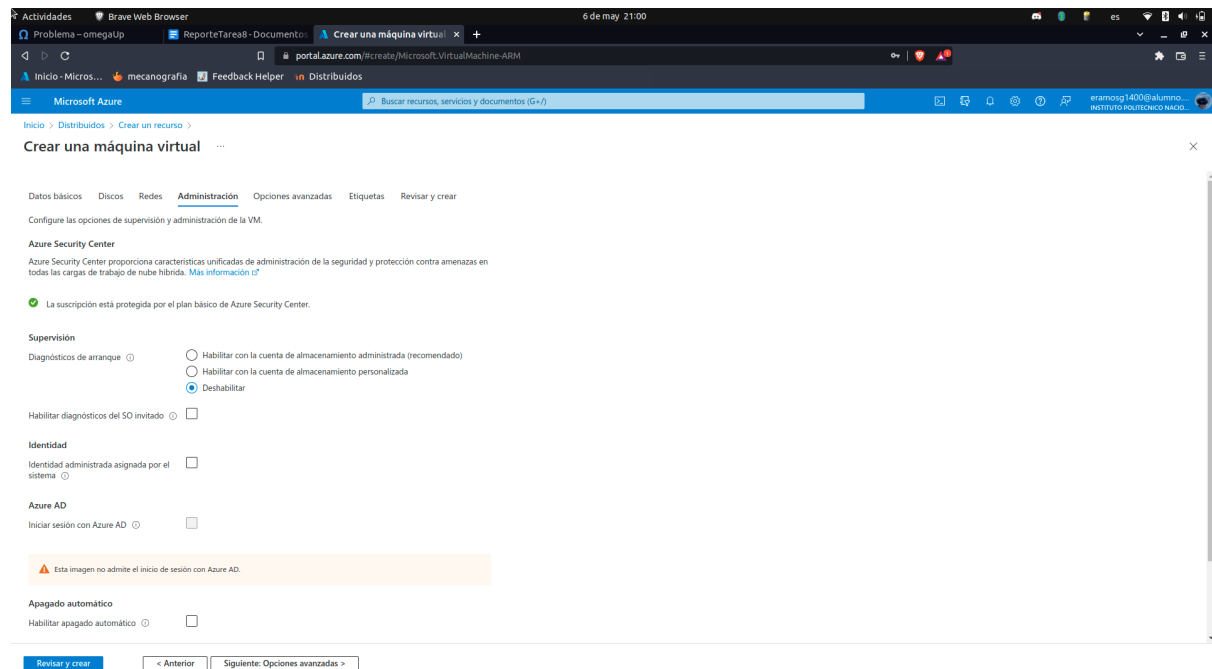
Crear y adjuntar un nuevo disco

Asociar un disco existente

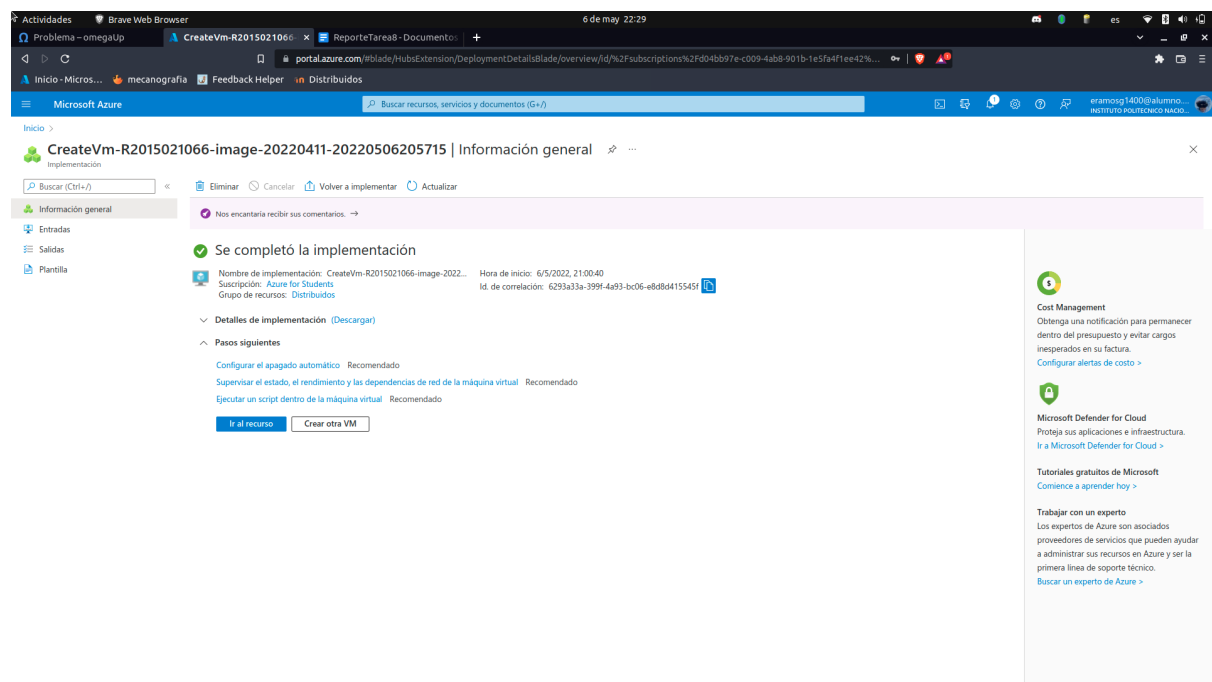
Revisar y crear

< Anterior

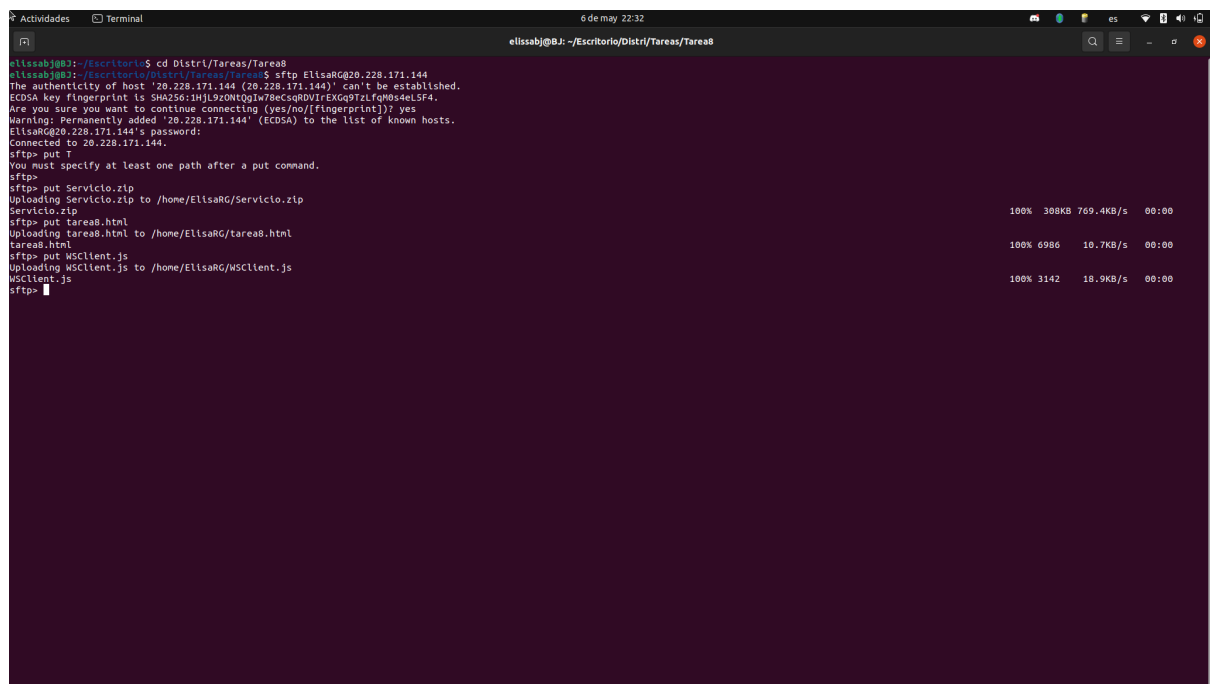
Siguiente: Redes >



Una vez finalizada la configuración como vimos anteriormente damos click en el botón Ir al recurso.



Subimos archivos a nuestra máquina virtual.

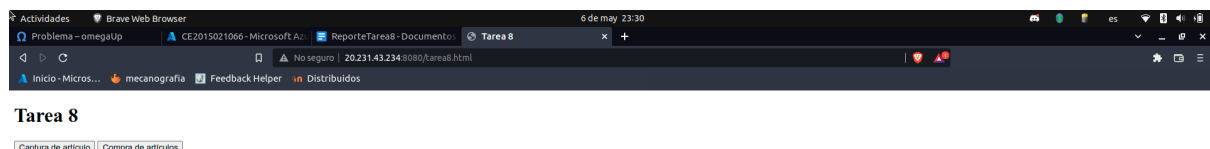


```
elissabj@B.J: ~/Escritorio/Distri/Tareas/Tarea8
elissabj@B.J: ~/Escritorio/Distri/Tareas/Tarea8$ cd Distri/Tareas/Tarea8
elissabj@B.J: ~/Escritorio/Distri/Tareas/Tarea8$ sftp ElisaRG@20.228.171.144
The authenticity of host '20.228.171.144 (20.228.171.144)' can't be established.
ECDSA key fingerprint is SHA256:1HJL9Z0NTQg1w78eCsqRDVrEXGq9TzLfQM0s4eLSF4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '20.228.171.144' (ECDSA) to the list of known hosts.
ElisaRG@20.228.171.144's password:
Connected to 20.228.171.144.
sftp> put ?
You must specify at least one path after a put command.
sftp>
sftp> put Servicio.zip
Uploading Servicio.zip to /home/ElisaRG/Servicio.zip
Servicio.zip                                100% 308KB 769.4KB/s 00:00
sftp> put tarea8.html
Uploading tarea8.html to /home/ElisaRG/tarea8.html
tarea8.html                                100% 6986 10.7KB/s 00:00
sftp> put WSClient.js
Uploading WSClient.js to /home/ElisaRG/WSClient.js
WSClient.js                                100% 3142 18.9KB/s 00:00
sftp>
```

## Pruebas

Para las pruebas se inicializa el tomcat como en las tareas anteriores no hay que olvidar que ahora debemos de cargar el nuevo:

1. El sistema desplegará inicialmente un menú con dos botones donde se podrá seleccionar las siguientes opciones: Captura de artículo y Compra de artículos.





2. Al seleccionar la opción "Captura de artículo" el sistema desplegará la pantalla "Captura de artículos" la cual permitirá capturar la descripción del artículo, el precio, la cantidad en almacén y la fotografía del artículo. Los datos de los artículos se guardarán en una tabla llamada "articulos". Cada artículo tendrá un ID autoincremental.

The screenshot shows a Brave Web Browser window with the URL `20.231.43.234:8080/tarea8.html`. The page title is "Captura de artículo". The form contains the following fields:

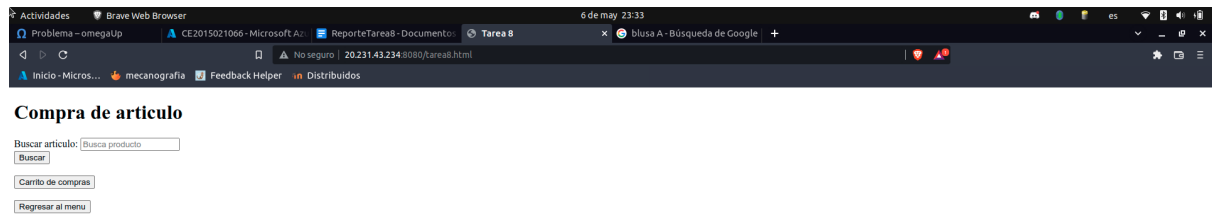
- 
- 
- 

Below the form is a section for file upload with the text "Elegir archivos" and "Sin archivos seleccionados". There are two buttons: "Registrar artículo" and "Regresar al menú".

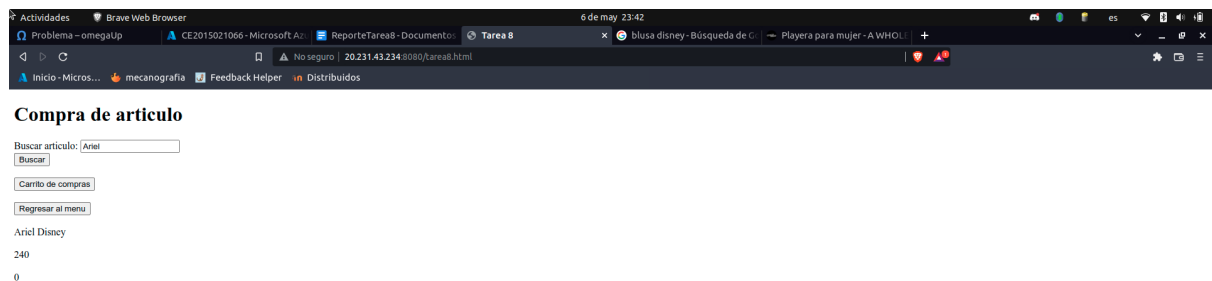
A dark overlay message box is displayed in the center-right of the screen, containing the text: "20.231.43.234:8080 dice" and "El artículo se dio de alta". A red "Aceptar" button is at the bottom right of the message box.

```
ElisaRG@CE2015021066: ~  
ElisaRG@CE2015021066: ~$ cp WSClient.js ~/apache-tomcat-8.5.78/webapps/ROOT/  
ElisaRG@CE2015021066: ~$ cp Tarea8.html ~/apache-tomcat-8.5.78/webapps/ROOT/  
ElisaRG@CE2015021066: ~$ mysql -u hugo -p  
Enter password:  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 5  
Server version: 5.7.38-0ubuntu0.18.04.1 (Ubuntu)  
Copyright (c) 2000, 2022, 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> select * from articulo;  
ERROR 1146 (42S02): Table 'servicio_web.articulo' doesn't exist  
mysql> select * from articulos;  
+----+-----+-----+-----+  
| id_articulo | description | precio | cantidad | foto |  
+----+-----+-----+-----+  
| 1 | Blusa A | 360 | 8 | NULL |  
+----+-----+-----+-----+  
1 row in set (0.00 sec)  
mysql>
```

- Al seleccionar la opción "Compra de artículos" el sistema desplegará la pantalla "Compra de artículos" la cual permitirá al usuario buscar artículos ingresando una palabra la cual se buscará en el campo "descripcion" de la tabla "articulos". La búsqueda se realizará utilizando la cláusula LIKE de MySQL.



- Los datos de los artículos (fotografía, descripción y precio) que resulten de una búsqueda se desplegarán en la pantalla "Compra de artículos".



## Conclusiones.

Elisa Ramos Gomez: La tarea está muy laboriosa, la verdad apenas comencé comprendiendo y experimentando más en los servicios web, ya que en lo personal no es mi fuerte, de igual manera reconozco que esta ocasión fusione de actividades dejándome con poco tiempo disponible para poder realizarla, ya que subestime el tiempo de desarrollo, pero al menos lo que se pudo implementar en el tiempo reducido que le dedique; me tiene reflexionando en mis áreas de oportunidad como futura ingeniera, donde mi principal propósito es terminar lo que ya se empezó para poder continuar con las demás responsabilidades, planeando situaciones de emergencia para cuando me vuelva a ver esta situación; por el lado técnico creo que gracias a mi trabajo pude desarrollar (con tiempo limitado) los resultados que presento, la opción que tomé de irlo probando de manera virtual por cada uno de los cambios que realizaba, reconozco que no fue la mejor metodología de trabajo; de igual forma profundizar en los nuevos lenguajes de programación como JavaScript ya que al tratar de modificar el WSCliente.js que se había proveído me ayudó más cuando lo modificaba obtenía errores que tuvo poco éxito en resolver por el momento, al igual que las pocas optimizaciones que uso al implementar el cliente ya que en el DOM lo que hago es ir ingresando hijos lo cual lo vuelve impráctico; de las peticiones de parte del back-end seguí los ejemplos mostrados en clase para poder implementar las peticiones mínimas que se pedían.