



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Tarea 8. Desarrollo de un sistema utilizando un servicio web

Unidad de aprendizaje: Desarrollo de Sistemas Distribuidos

Grupo: 4CV11

Alumno:
Sanchez Mendez Edmundo Josue

Profesor:
Pineda Guerrero Carlos

Índice

1. Introducción	2
2. Desarrollo	3
2.1. Creación de la maquina virtual	3
2.2. Preparación del entorno y creación de la base de datos	8
2.3. Compilación del código para el back-end y colocación de archivos para el front-end en ROOT	11
3. Pruebas	12
3.1. Prueba 1. Captura de articulo	12
3.2. Prueba 2. Búsqueda de artículos	24
3.3. Prueba 3. Compra de artículos	27
3.4. Prueba 4. Carrito de compra	39
4. Conclusiones	42
Referencias	42

1. Introducción

Desarrollar un prototipo de sistema de comercio electrónico utilizando un servicio web estilo REST escrito en lenguaje Java.

Requerimientos funcionales

1. El sistema desplegará inicialmente un menú 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 una pantalla que 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 auto-incremental.
3. Al seleccionar la opción “Compra de artículos” el sistema desplegará una pantalla que 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 SQL.
4. Los datos de los artículos (fotografía, descripción y precio) que resulten de una búsqueda se desplegarán en la pantalla “Compra de artículos”.
5. Para cada artículo resultado de la búsqueda, se desplegará un botón de “Compra” y un campo de “Cantidad” con un valor default igual a 1.
6. Cuando el usuario presione el botón de “Compra”, si la cantidad de artículos a comprar es menor o igual a la cantidad de artículos en la tabla “articulos”, se escribirá en una tabla llamada “carrito_compra” el ID del artículo y la cantidad, así mismo se restará la cantidad solicitada de la cantidad en la tabla de “artículos”, de otra manera se desplegará un mensaje indicando al usuario el número de artículos disponibles. La escritura a la tabla “carrito_compra” y la actualización de la tabla “artículos” se deberán realizar dentro de una transacción.
7. En la pantalla de “Compra de artículos” se dispondrá de un botón “Carrito de compra” el cual desplegará una pantalla con la lista de artículos en la tabla “carrito_compra”, incluyendo una pequeña foto del artículo, descripción del artículo, cantidad, precio y costo (cantidad x precio).
8. La pantalla de carrito de compra tendrá un botón que permitirá regresar a la pantalla “Compra de artículos”.

Requerimientos no funcionales

1. El back-end consistirá en un servicio web sobre Tomcat.
2. El back-end ejecutará en una máquina virtual con Ubuntu en Azure.
3. El DBMS será MySQL el cual ejecutará en la misma máquina virtual donde ejecutará Tomcat.
4. El front-end podrá ser desarrollado en cualquier lenguaje que permita ejecutarlo en un dispositivo móvil. Se recomienda utilizar HTML-Javascript.

2. Desarrollo

2.1. Creación de la maquina virtual

En esta parte veremos la creación de la maquina virtual en donde llevaremos acabo la practica. Recordar que en la practica anterior creamos una imagen de la maquina virtual utilizada, por lo que usaremos esa imagen para poder crear la maquina virtual que usaremos para esta practica.

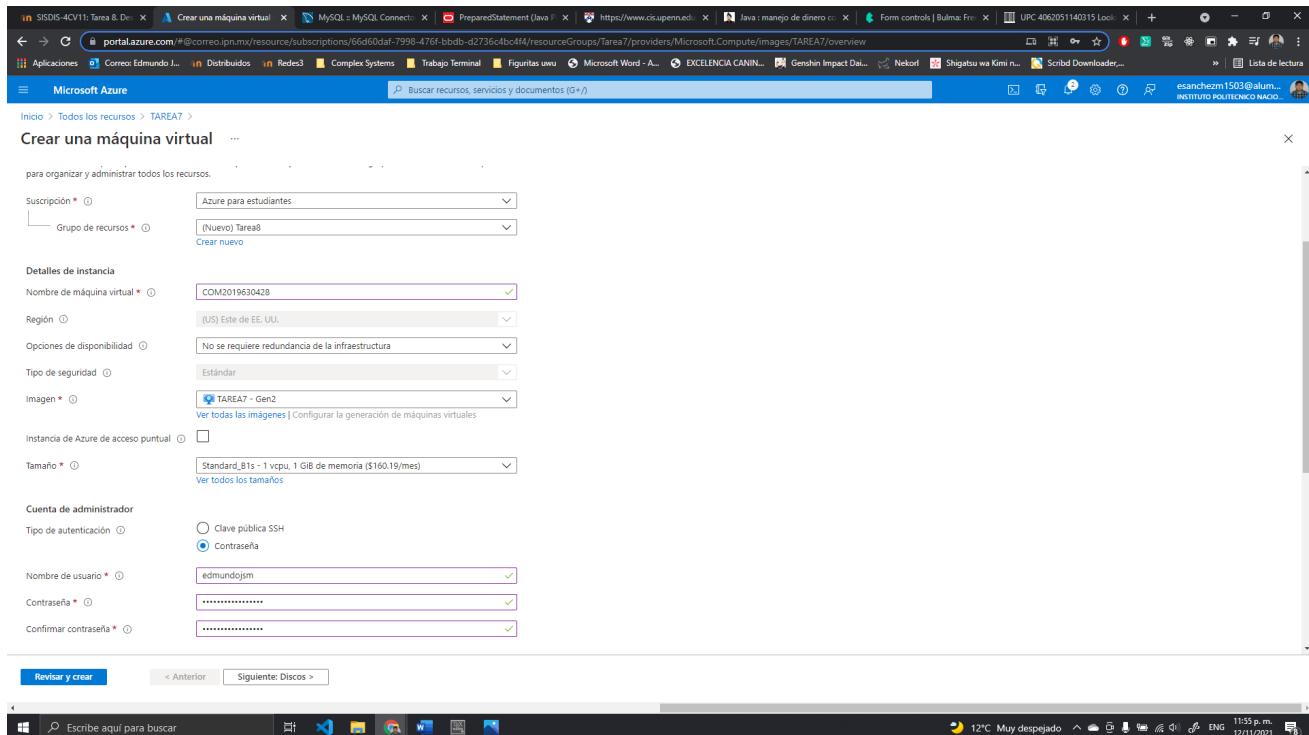


Figura 1: Datos básicos de la maquina virtual.

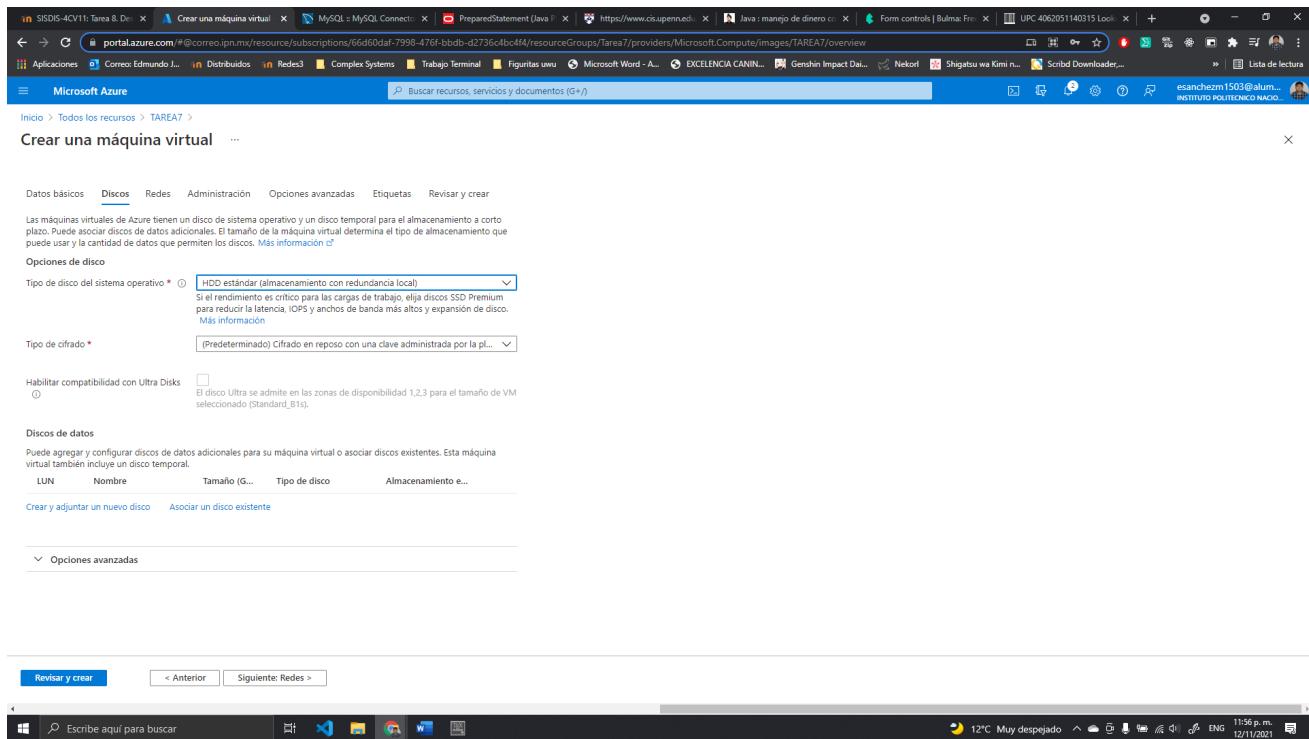


Figura 2: Configuración del tipo de disco de la maquina virtual.

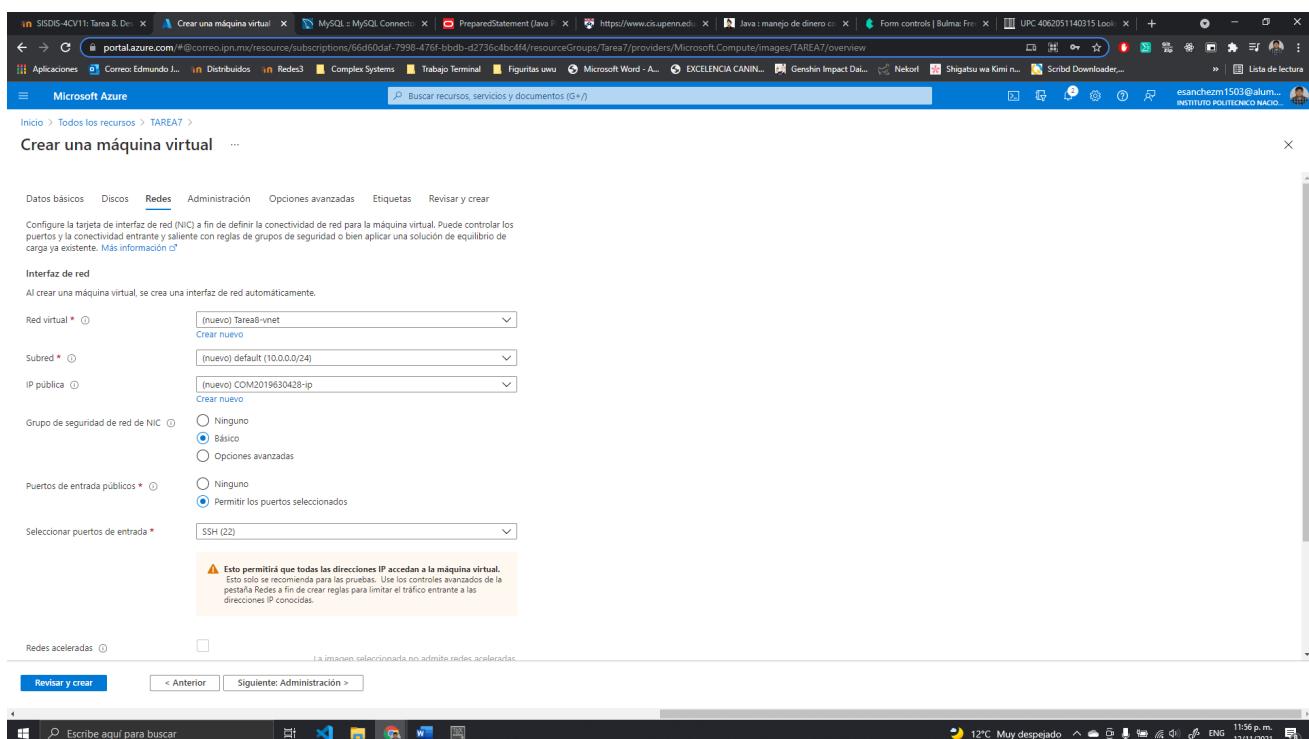


Figura 3: Información sobre la redes de la maquina virtual.

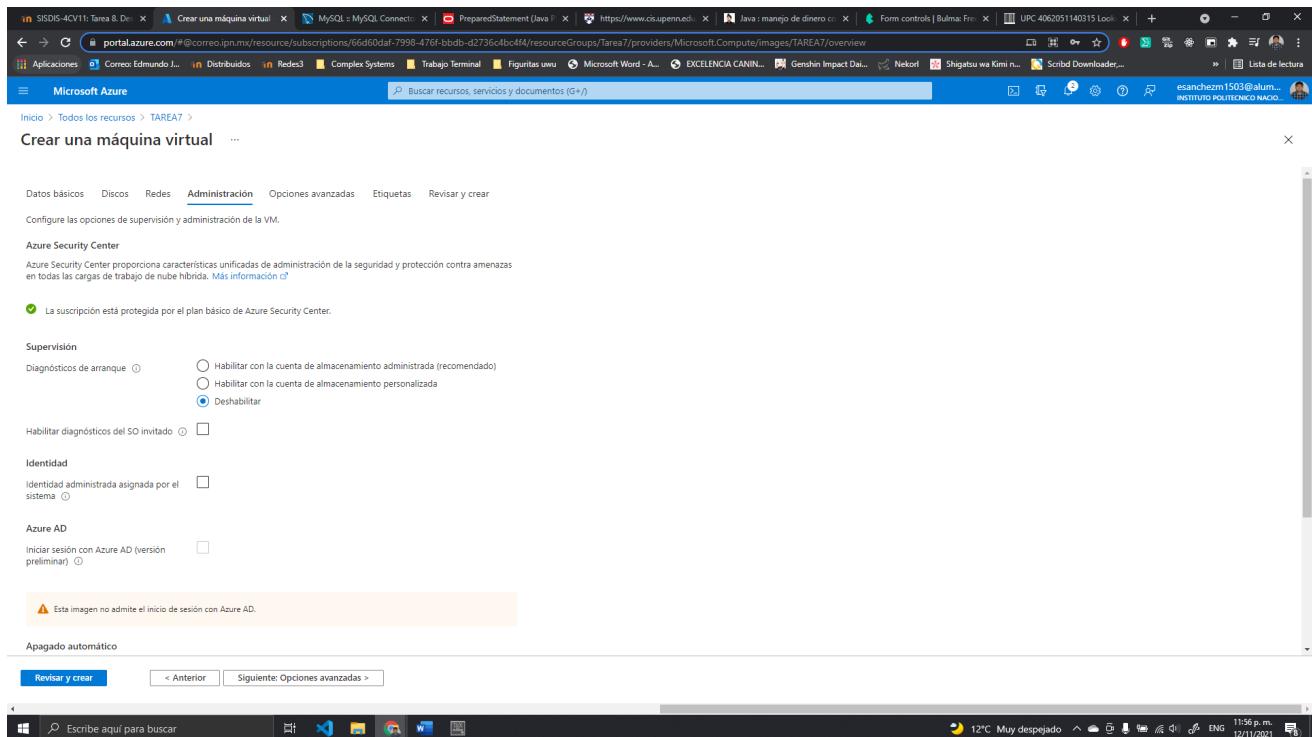


Figura 4: Configuración de la administración de la maquina virtual.

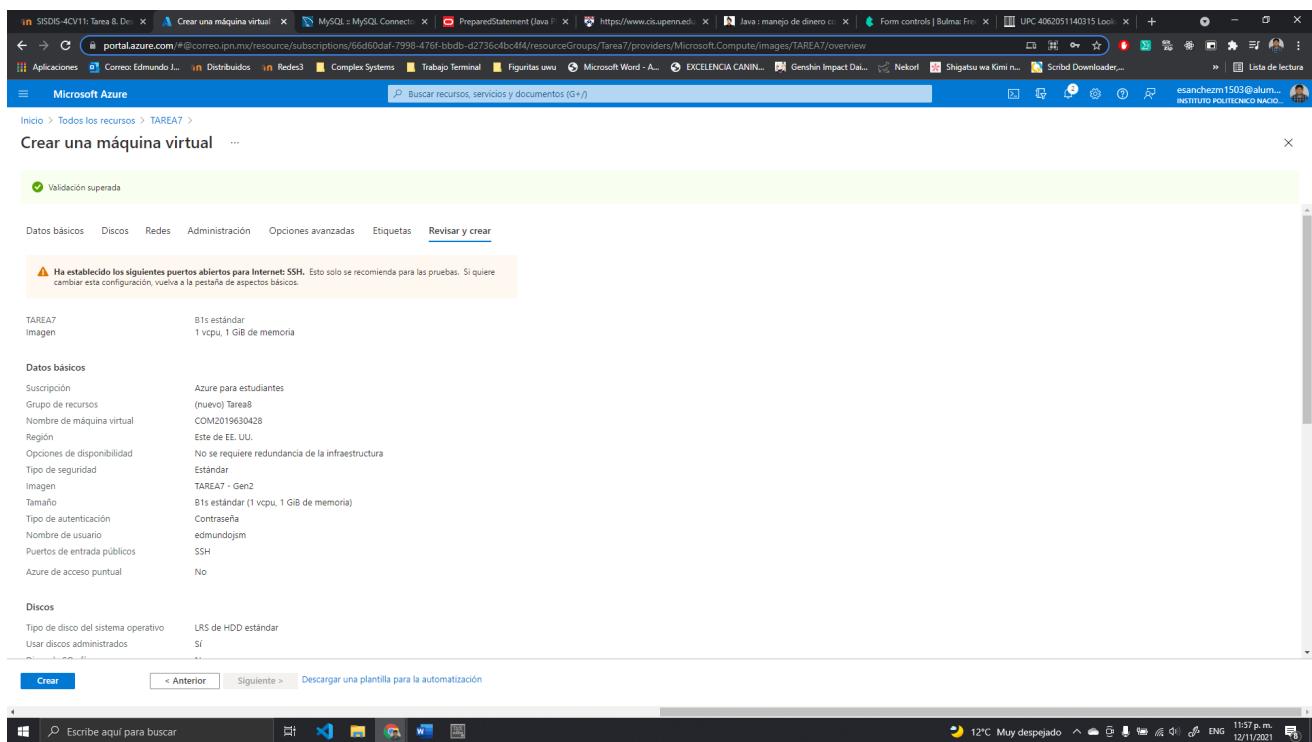


Figura 5: Creación de la maquina virtual.

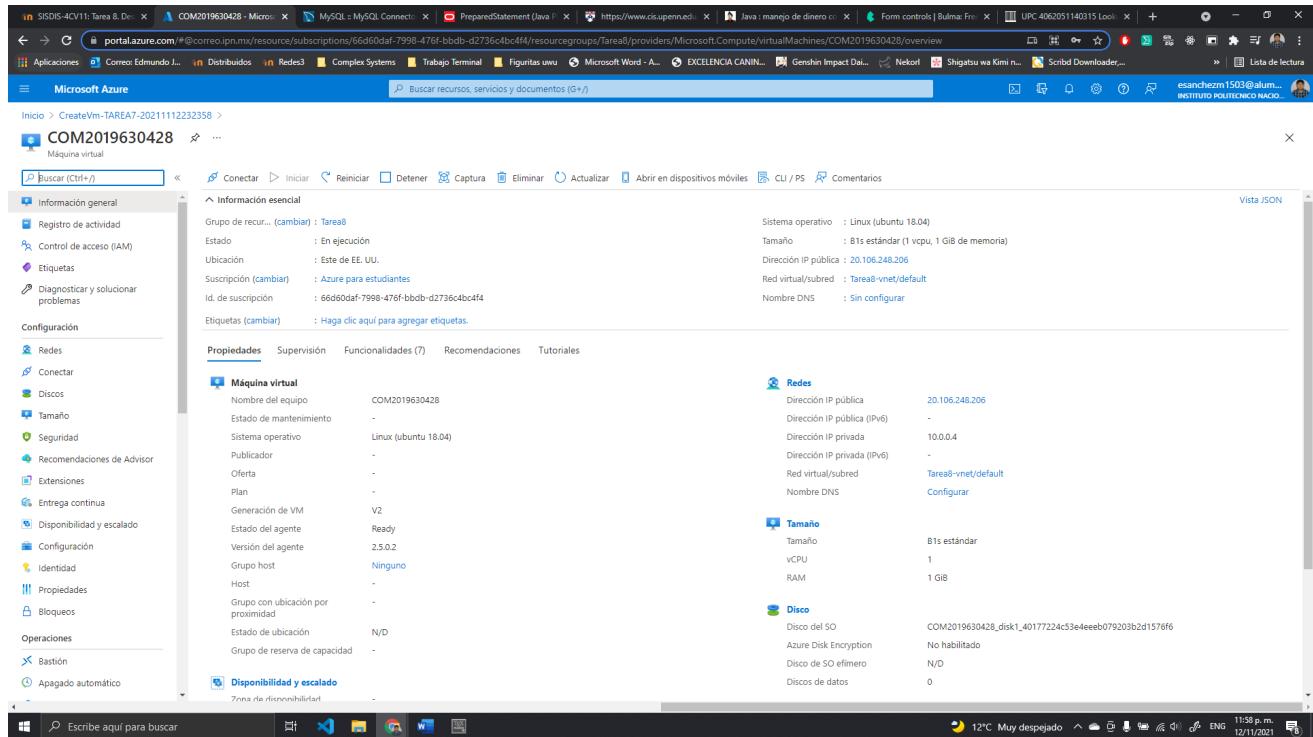


Figura 6: Panel de control de la maquina virtual.

Una vez creada la maquina virtual tenemos que abrir el puerto 8080, ya que este puerto es el que ocupa Apache Tomcat, así podemos conectarnos de manera remota desde cualquier dispositivo , también es importante mencionar que si no se configura de manera correcta el puerto 8080 la conexión con la maquina virtual para poder visualizar el servicio web. En las figuras 7 y 8 podemos ver la configuración del puerto.

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with various service icons like Applications, Distributions, Networks, etc. The main area is titled 'Redes' (Networks) under 'COM2019630428'. It shows a list of network interfaces, with 'com2019630428746' selected. In the center, there's a table of security rules for this interface. A modal window on the right is titled 'Agregar regla de seguridad de entrada' (Add incoming security rule) for 'puerto_8080'. The modal includes fields for Origin (Any), Destination (Any), Service (Custom), and Port (8080). The 'Protocolo' section has 'TCP' selected. The 'Acción' section has 'Permitir' selected. At the bottom of the modal are 'Agregar' (Add) and 'Cancelar' (Cancel) buttons.

Figura 7: Configuración del puerto 8080.

This screenshot shows the same Azure portal interface as Figure 7, but with the security rule added. The table now includes a new row for 'Puerto_8080' with a priority of 310, protocol TCP, and action 'Permitir'. The 'Agregar regla de puerto de entrada' button is visible at the top right of the table area.

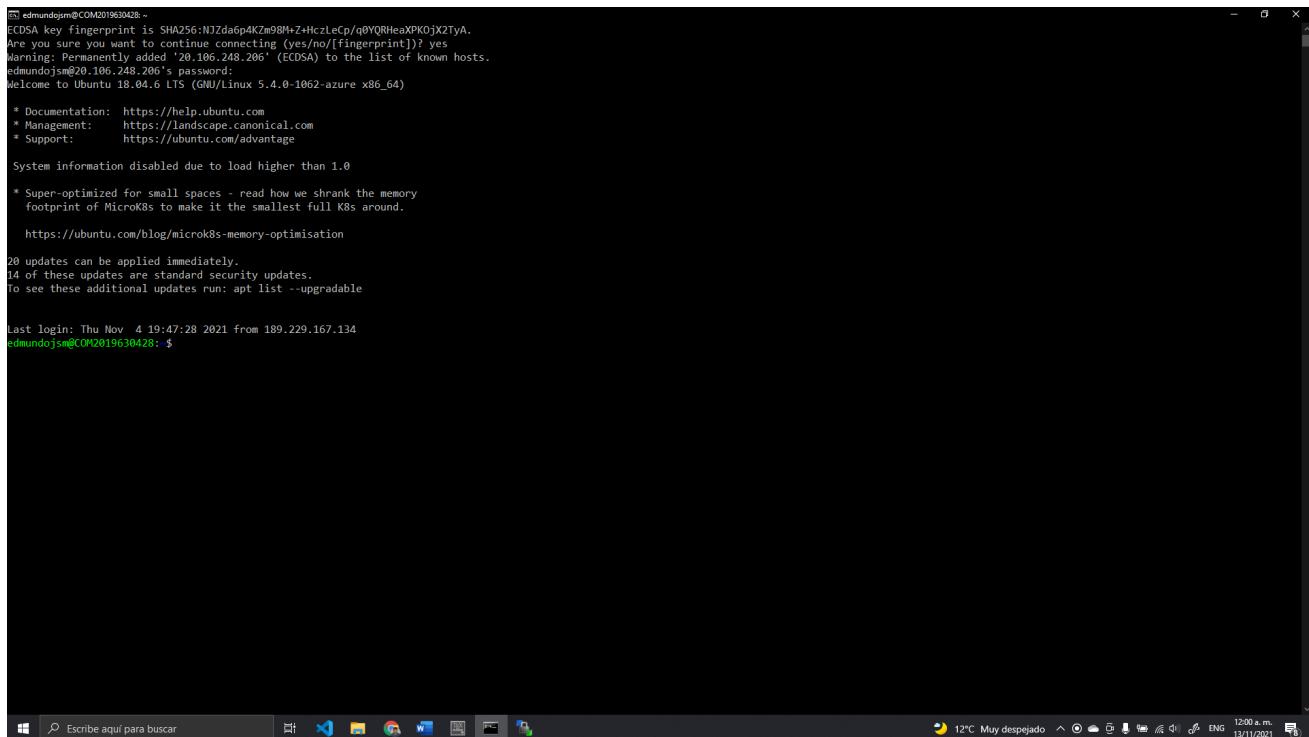
Figura 8: Puerto 8080 abierto correctamente.

Ya teniendo la maquina preparada, procederemos a compilar nuestra aplicación estilo REST la cual fue desarrollada en Java, y a colocar los archivos HTML, JavaScript e imagen que usaremos como parte del front-end de la aplicación, pero antes de ellos veamos la preparación del entorno para ejecutar el

sistema así como la creación de la base de datos a usar y la explicación del porque se añaden mas campos que los solicitados en la tarea.

2.2. Preparación del entorno y creación de la base de datos

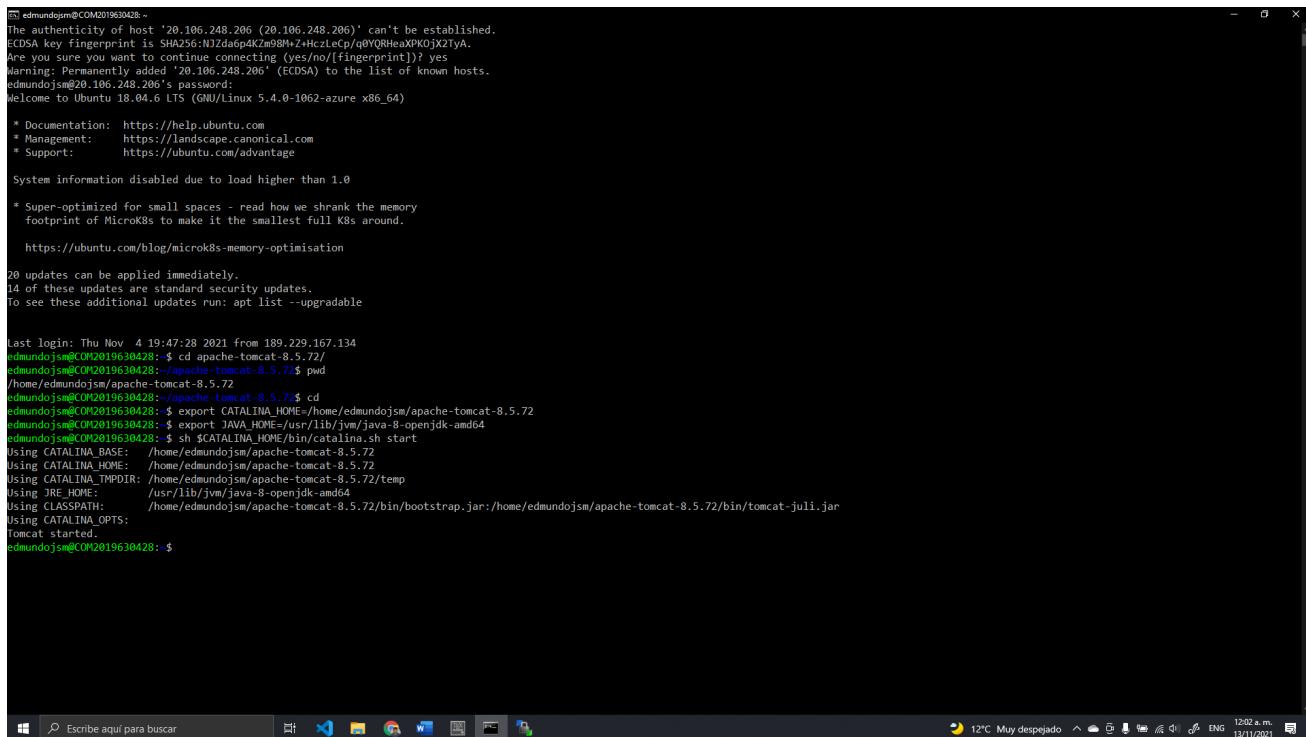
Para empezar debemos conectarnos a la maquina virtual vía SSH como podemos ver en la figura 9.



```
edmundojpm@COM2019630428:~  
ECDSA key fingerprint is SHA256:NJZdaop4KZm98M+Z+HczLieCp/q0YQRHeaXPK0jX2TyA.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '20.106.248.206' (ECDSA) to the list of known hosts.  
edmundojpm@20.106.248.206's password:  
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1062-azure x86_64)  
  
 * Documentation: https://help.ubuntu.com  
 * Management: https://landscape.canonical.com  
 * Support: https://ubuntu.com/advantage  
  
System information disabled due to load higher than 1.0  
  
* Super-optimized for small spaces - read how we shrank the memory  
 footprint of MicroK8s to make it the smallest full K8s around.  
  
https://ubuntu.com/blog/microk8s-memory-optimisation  
  
20 updates can be applied immediately.  
14 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
  
Last login: Thu Nov  4 19:47:28 2021 from 189.229.167.134  
edmundojpm@COM2019630428:~$
```

Figura 9: Conexión SSH con la maquina virtual.

Ahora necesitaremos crear nuestras variables de entorno de CATLINA_HOME y JAVA_HOME así como hacer que Tomcat empiece a funcionar, como lo vemos en la figura 10.



```

edmundojsm@COM2019630428 ~
The authenticity of host '20.106.248.206 (20.106.248.206)' can't be established.
ECDSA key fingerprint is SHA256:NJ7da6ndKZnB8MzHczLcP/qYOR8heaxPK0jX2TyA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '20.106.248.206' (ECDSA) to the list of known hosts.
edmundojsm@20.106.248.206's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1062-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information disabled due to load higher than 1.0

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

20 updates can be applied immediately.
14 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Thu Nov  4 19:47:28 2021 from 189.229.167.134
edmundojsm@COM2019630428:~$ cd apache-tomcat-8.5.72/
edmundojsm@COM2019630428:~/apache-tomcat-8.5.72$ pwd
/home/edmundojsm/apache-tomcat-8.5.72
edmundojsm@COM2019630428:~/apache-tomcat-8.5.72$ cd
edmundojsm@COM2019630428:~$ export CATALINA_HOME=/home/edmundojsm/apache-tomcat-8.5.72
edmundojsm@COM2019630428:~$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
edmundojsm@COM2019630428:~$ $CATALINA_HOME/bin/catalina.sh start
Using CATALINA_BASE: /home/edmundojsm/apache-tomcat-8.5.72
Using CATALINA_HOME: /home/edmundojsm/apache-tomcat-8.5.72
Using CATALINA_TMPDIR: /home/edmundojsm/apache-tomcat-8.5.72/temp
Using JRE_HOME: /usr/lib/jvm/java-8-openjdk-amd64
Using CLASPATH: /home/edmundojsm/apache-tomcat-8.5.72/bin/bootstrap.jar:/home/edmundojsm/apache-tomcat-8.5.72/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
edmundojsm@COM2019630428:~$
```

Figura 10: Variables de entorno asignadas y Tomcat corriendo.

Finalmente nos toca crear la base de datos a usar la cual esta basada con los requerimientos funcionales de la tarea, sin embargo, se añadieron dos campos mas los cuales considero importantes para un tipo de sistema como el que se esta desarrollando, estos campos son nombre de tipo varchar y UPC de tipo bigint, este ultimo es conocido como CÓDIGO UNIVERSAL DEL PRODUCTO, el cual es usado para poder rastrear los artículos comerciales en las tiendas, este UPC es el clásico código de barras que podemos encontrar en los artículos que compramos, esto nos podría permitir tener una mejor gestión en nuestro almacén tanto física como de manera virtual, esto claro, si se desea hacer aun mas grande esta aplicación, la razón del por cual es de tipo bigint es porque consta de 12 dígitos numéricos y bigint es el único tipo de dato que soporta la longitud de este dato, mencionar que por obvias razones un producto tiene un único UPC, mencionar que en las pruebas el UPC ingresado son números menores a 12 dígitos esto debido para facilitar la visualización de la información. Finalmente el script de la creación de tablas y de las llaves foráneas se anexan en un archivo de texto con el nombre de “Base de datos”. Todo el proceso de creación lo podemos ver en las figuras 11 y 12.

```
ed@edmundopgm:~/COM2019630428-$
ed@edmundopgm:~/COM2019630428-$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 5.7.36-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2021, 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> grant all on tienda_web.* to hugo@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> quit
Bye
ed@edmundopgm:~/COM2019630428-$ mysql -u hugo -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.7.36-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2021, 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> create database tienda_web;
Query OK, 1 row affected (0.00 sec)

mysql> use tienda_web;
Database changed
mysql> create table articulos
    > (
    >     id_articulo integer auto_increment primary key,
    >     nombre varchar(50) not null,
    >     descripcion varchar(256) not null,
    >     UPC bigint not null,
    >     precio dec(8,2) not null,
    >     cantidad_almacen integer not null
    > );
Query OK, 0 rows affected (0.07 sec)
```

Figura 11: Dando permisos al usuario hugo para la base de datos a usar y primera parte de la creación de esta.

```
edmundopm@COM2019630428:~$ mysql -u root -p
mysql> create table fotos_articulos
    > (
    >     id_foto integer auto_increment primary key,
    >     foto longblob,
    >     id_articulo integer not null
    > );
Query OK, 0 rows affected (0.08 sec)

mysql> create table carrito_compra
    > (
    >     id_compra integer auto_increment primary key,
    >     cantidad integer not null,
    >     id_articulo integer not null
    > );
Query OK, 0 rows affected (0.08 sec)

mysql> alter table fotos_articulos add foreign key (id_articulo) references articulos(id_articulo);
Query OK, 0 rows affected (0.20 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table carrito_compra add foreign key (id_articulo) references articulos(id_articulo);
Query OK, 0 rows affected (0.19 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show tables;
+---------------------+
| Tables_in_tienda_web |
+-----+
| articulos           |
| carrito_compra      |
| fotos_articulos    |
+-----+
3 rows in set (0.00 sec)

mysql> quit
Bye
edmundopm@COM2019630428:~$
```

Figura 12: Ultima parte de la creación de la base de datos.

2.3. Compilación del código para el back-end y colocación de archivos para el front-end en ROOT

Ahora tome como base el código de servicio, solo cambiamos la clase Usuario por una clase Artículo y modificar el archivo Servicio.java para cumplir con los requerimientos funcionales, mencionar también que como tal no hay compilación del código para el front-end ya que el front-end fue desarrollado con HTML-JS y este no necesita compilación pero si necesitamos tener los archivos en el servidor Tomcat en la carpeta ROOT. Ahora nos toca compilar el servicio, generar el .war y copiarlo a la carpeta webapps para poder tener el servicio en linea usando Tomcat, todo esto se ve en la figura 13 y 14, finalmente vemos los contenidos en la carpeta webapps y ROOT de nuestra carpeta de Tomcat, esto también lo podemos ver en la figura 14.

```

edmundojsm@COM2019630428:~/Servicio
edmundojsm@COM2019630428:~/Servicio/negocio$ ls -l
total 56
-rw-rw-r-- 1 edmundojsm edmundojsm 1799 Oct 24 20:51 AdaptadorGsonBase64.class
-rw-rw-r-- 1 edmundojsm edmundojsm 870 Oct 19 08:51 AdaptadorGsonBase64.java
-rw-rw-r-- 1 edmundojsm edmundojsm 585 Nov 12 01:21 Artículo.java
-rw-rw-r-- 1 edmundojsm edmundojsm 278 Oct 24 20:01 Error.class
-rw-rw-r-- 1 edmundojsm edmundojsm 245 Oct 19 09:00 Error.java
-rw-rw-r-- 1 edmundojsm edmundojsm 139 Oct 19 09:00 Foto.java
-rw-rw-r-- 1 edmundojsm edmundojsm 8191 Nov 4 18:47 Servicio.class
-rw-rw-r-- 1 edmundojsm edmundojsm 12456 Nov 13 04:32 Servicio.java
-rw-rw-r-- 1 edmundojsm edmundojsm 924 Nov 4 18:47 Usuario.class
-rw-rw-r-- 1 edmundojsm edmundojsm 604 Nov 4 18:23 Usuario.java
edmundojsm@COM2019630428:~/Servicio/negocio$ cd ..
edmundojsm@COM2019630428:~/Servicio$ javac -cp $CATALINA_HOME/lib/javax.ws.rs-api-2.0.1.jar:$CATALINA_HOME/lib/gson-2.3.1.jar:. negocio/Servicio.java
edmundojsm@COM2019630428:~/Servicio$ cp negocio/* WEB-INF/classes/negocio/*
edmundojsm@COM2019630428:~/Servicio$ jar cvf Servicio.war WEB-INF META-INF
added manifest
adding: WEB-INF/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/web.xml(in = 672) (out= 296)(deflated 55%)
adding: WEB-INF/classes/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/negocio/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/negocio/Usuario.class(in = 924) (out= 532)(deflated 42%)
adding: WEB-INF/classes/negocio/Artículo.class(in = 841) (out= 492)(deflated 41%)
adding: WEB-INF/classes/negocio/AdaptadorGsonBase64.class(in = 1799) (out= 737)(deflated 59%)
adding: WEB-INF/classes/negocio/Servicio.class(in = 9536) (out= 4422)(deflated 53%)
adding: WEB-INF/classes/negocio/Error.class(in = 278) (out= 214)(deflated 23%)
ignoring entry META-INF/
adding: META-INF/context.xml(in = 304) (out= 210)(deflated 30%)
edmundojsm@COM2019630428:~/Servicio$
```

Figura 13: Compilación de Servicio.

```
edmundojsm@COM2019630428:~/apache-tomcat-8.5.72/webapps/ROOT
edmundojsm@COM2019630428:~/apache-tomcat-8.5.72/webapps$ cd
edmundojsm@COM2019630428:~$ cd apache-tomcat-8.5.72/
edmundojsm@COM2019630428:~/apache-tomcat-8.5.72$ cd webapps/
edmundojsm@COM2019630428:~/apache-tomcat-8.5.72/webapps$ ls -l
total 4
drwxrwxr-x 2 edmundojsm edmundojsm 4096 Nov 13 06:18 ROOT
edmundojsm@COM2019630428:~/apache-tomcat-8.5.72/webapps$ ls -l
total 20
drwxrwxr-x 2 edmundojsm edmundojsm 4096 Nov 13 06:18 ROOT
drwxr-x--- 4 edmundojsm edmundojsm 4096 Nov 13 06:21 Servicio
-rw-rw-r-- 1 edmundojsm edmundojsm 8671 Nov 13 06:20 Servicio.war
edmundojsm@COM2019630428:~/apache-tomcat-8.5.72/webapps$ cd ROOT
edmundojsm@COM2019630428:~/apache-tomcat-8.5.72/webapps/ROOT$ ls -l
total 44
-rw-rw-r-- 1 edmundojsm edmundojsm 3142 Nov 10 20:59 WsClient.js
-rw-rw-r-- 1 edmundojsm edmundojsm 20025 Nov 13 04:08 index.html
-rw-rw-r-- 1 edmundojsm edmundojsm 2907 Nov 11 04:46 nofoto.jpg
-rw-rw-r-- 1 edmundojsm edmundojsm 9181 Oct 24 20:08 prueba.html
-rw-rw-r-- 1 edmundojsm edmundojsm 1662 Oct 24 20:06 usuario_sin_foto.png
edmundojsm@COM2019630428:~/apache-tomcat-8.5.72/webapps/ROOT$
```

Figura 14: Contenidos de la carpeta webapps y ROOT de Tomcat.

3. Pruebas

Es importante mencionar que para el desarrollo de esta practica se usaron dos frameworks uno es Bulma el cual es un framework de CSS y SweetAlert2 el cual es un framework para alertas de JavaScript. Ahora pasemos a las pruebas del sistema desarrollado, mencionar que las pruebas se desarrollaron tanto en mi computadora personal como en mi celular para cumplir con uno de los requerimientos no funcionales y estas están vinculadas entre si.

3.1. Prueba 1. Captura de articulo

En los requerimientos funcionales nos solicitan mostrar 2 un menú con las opciones de Captura de articulo y Compra de artículos, sin embargo, decidí agregar dos mas, la opción de Inicio la cual nos mostrara los todos los artículos que se hayan registrados y en el caso de no tener ninguno nos dirá que no hay artículos y el de Carrito de compra el cual es solicitado que se nos presente al ver la opción de Compra de artículos, pero considere mas cómodo que se viera en todo momento para el usuario ya que considero que es mas amigable, esto lo podemos ver en la figura 15.

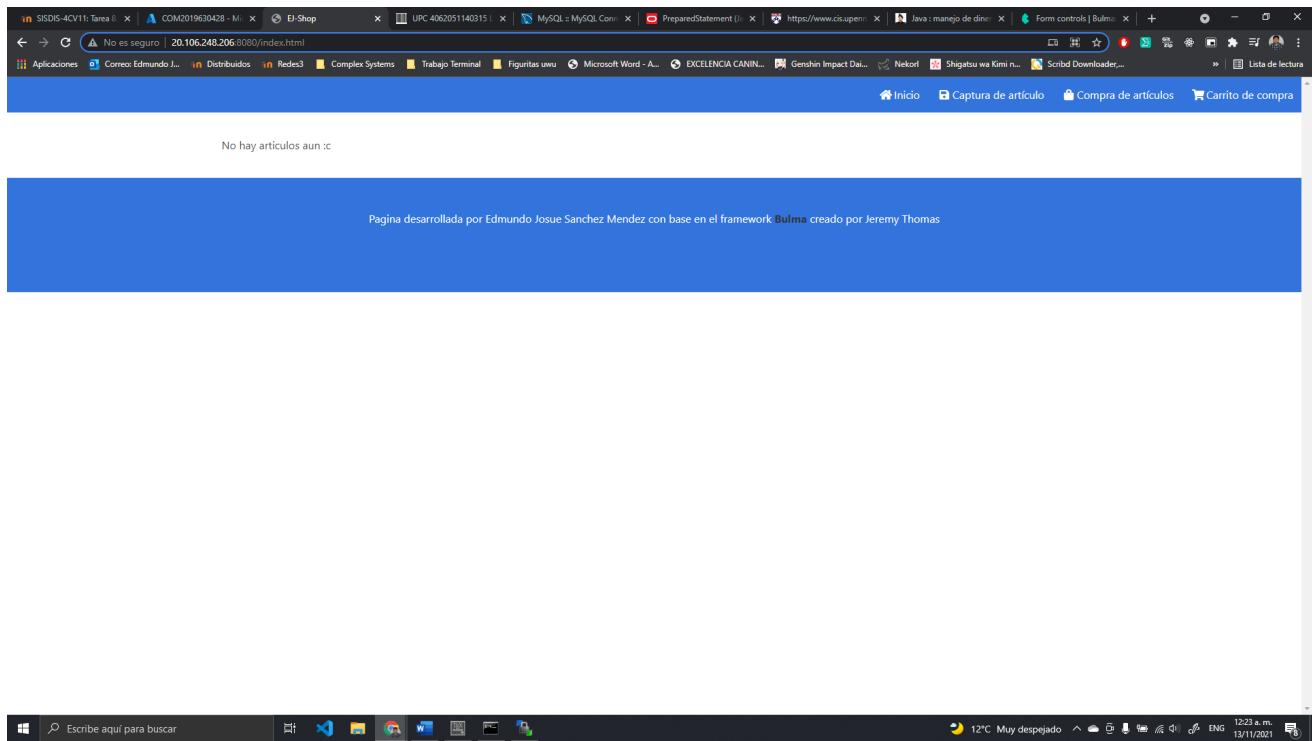


Figura 15: Pagina de inicio sin artículos registrados.

En la figura 16 podemos ver la captura de un artículo con toda la información que necesitamos guardar en la base de datos, esta prueba contiene una foto para al artículo, al darle click al botón de Guardar artículo y esperar pocos segundos nos sale la alerta que podemos ver en la figura 17, la cual indica como el artículo fue dado de alta correctamente y nos vacia el formulario incluyendo la imagen.

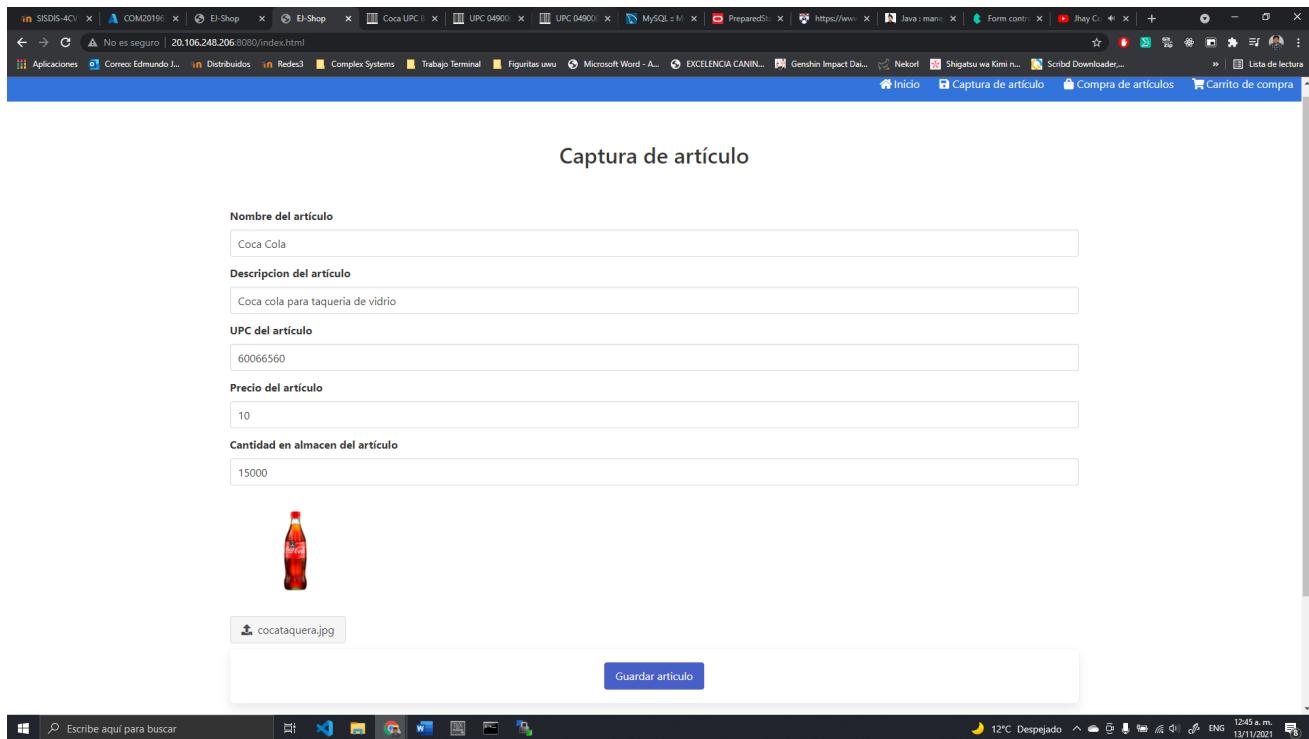


Figura 16: Prueba 1.1. Artículo dado de alta con foto.

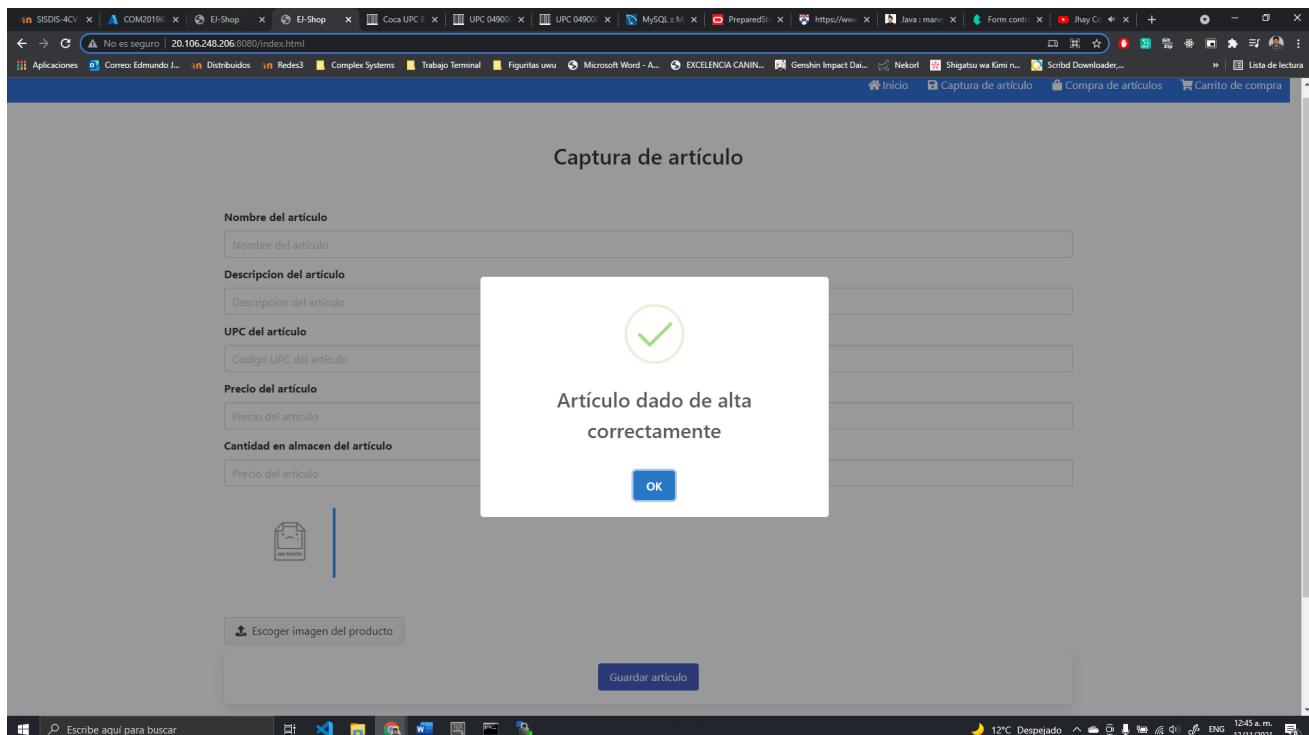


Figura 17: Resultado 1.1. Artículo dado de alta exitosamente con foto.

La siguiente prueba fue registrar un artículo sin foto como vemos en la figura 18 y en la figura 19 vemos como este fue dado de alta correctamente.

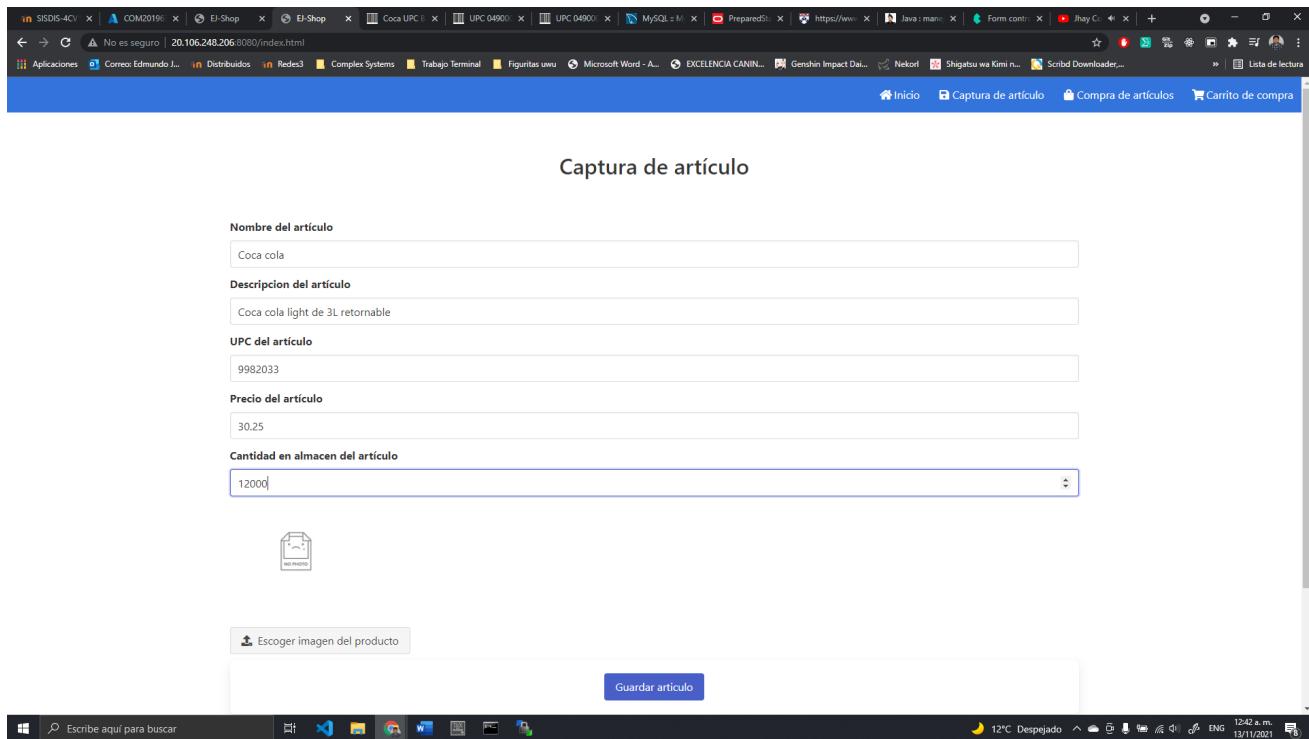


Figura 18: Prueba 1.2. Artículo dado de alta sin foto.

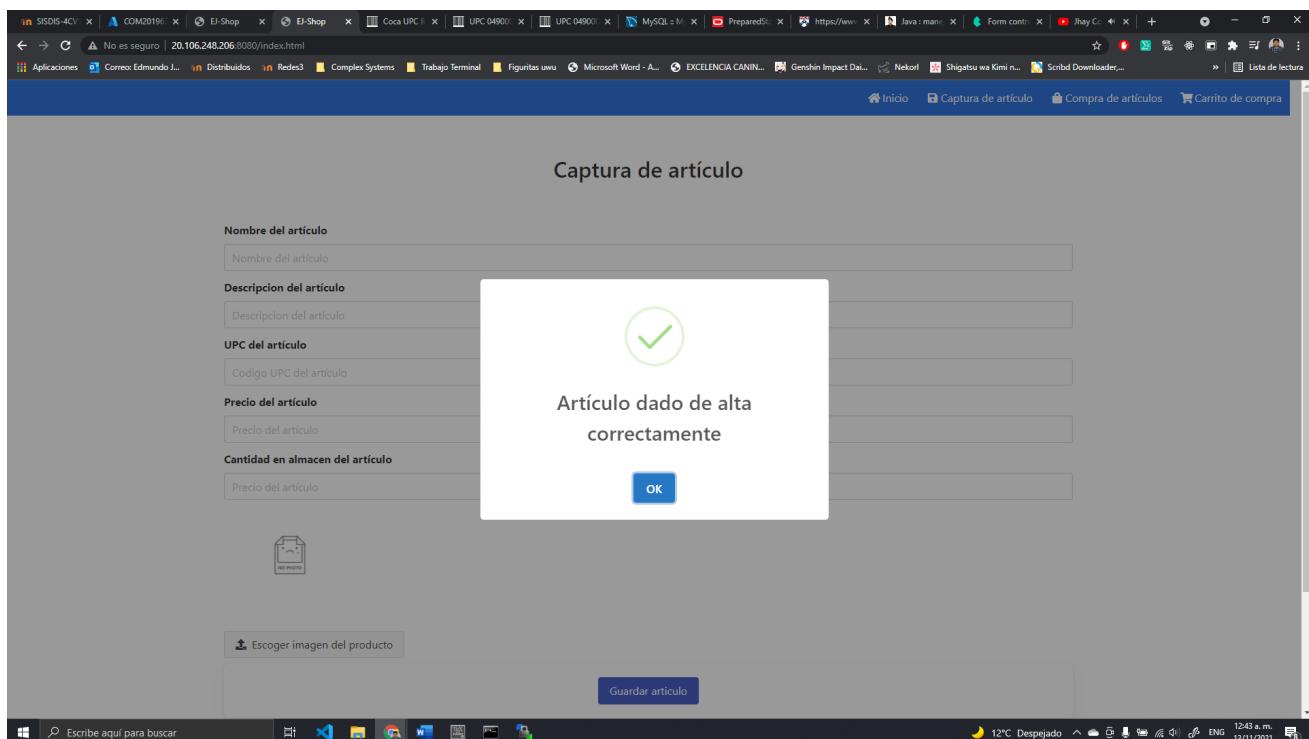


Figura 19: Resultado 1.2. Artículo dado de alta exitosamente sin foto.

La siguiente prueba fue ver que sucede si ponemos un UPC ya registrado con anterioridad y como vemos nos arroja un error de que el UPC ya existe, esto lo podemos ver en la figura 20, mencionar que este producto fue guardado en la base de datos pero evidentemente usando un UPC inexistente.

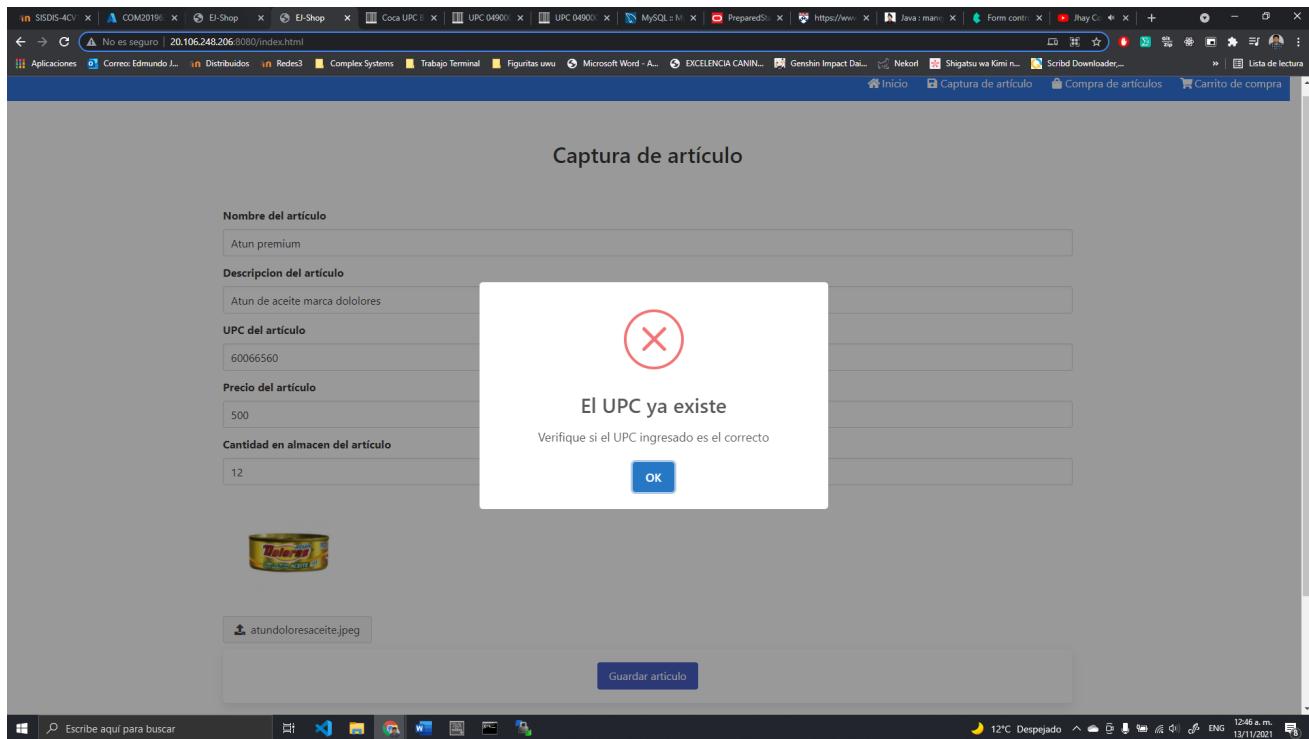
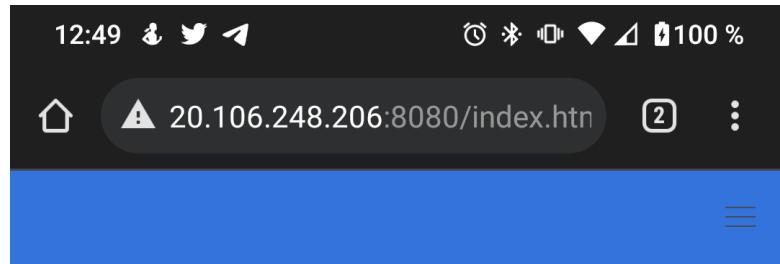


Figura 20: Resultado 1.3. Artículo con UPC ya guardo anteriormente.

Como ultima prueba se realizo un registro desde el celular, esto lo podemos ver en las figuras 21 y 22, añadir que en la figura 21 podemos ver en la parte superior una famosa burger navbaren la que si damos click se nos despliega las opciones que mencionamos al inicio. Finalmente en la figura 23 podemos ver como el articulo fue dado de alta de forma correcta



Captura de artículo

Nombre del artículo

Atun premium

Descripción del artículo

Atun de agua marca dolores

UPC del artículo

156304

Precio del artículo

500

Cantidad en almacen del artículo

10



Figura 21: Prueba 1.4. Artículo dado de alta con foto en celular.

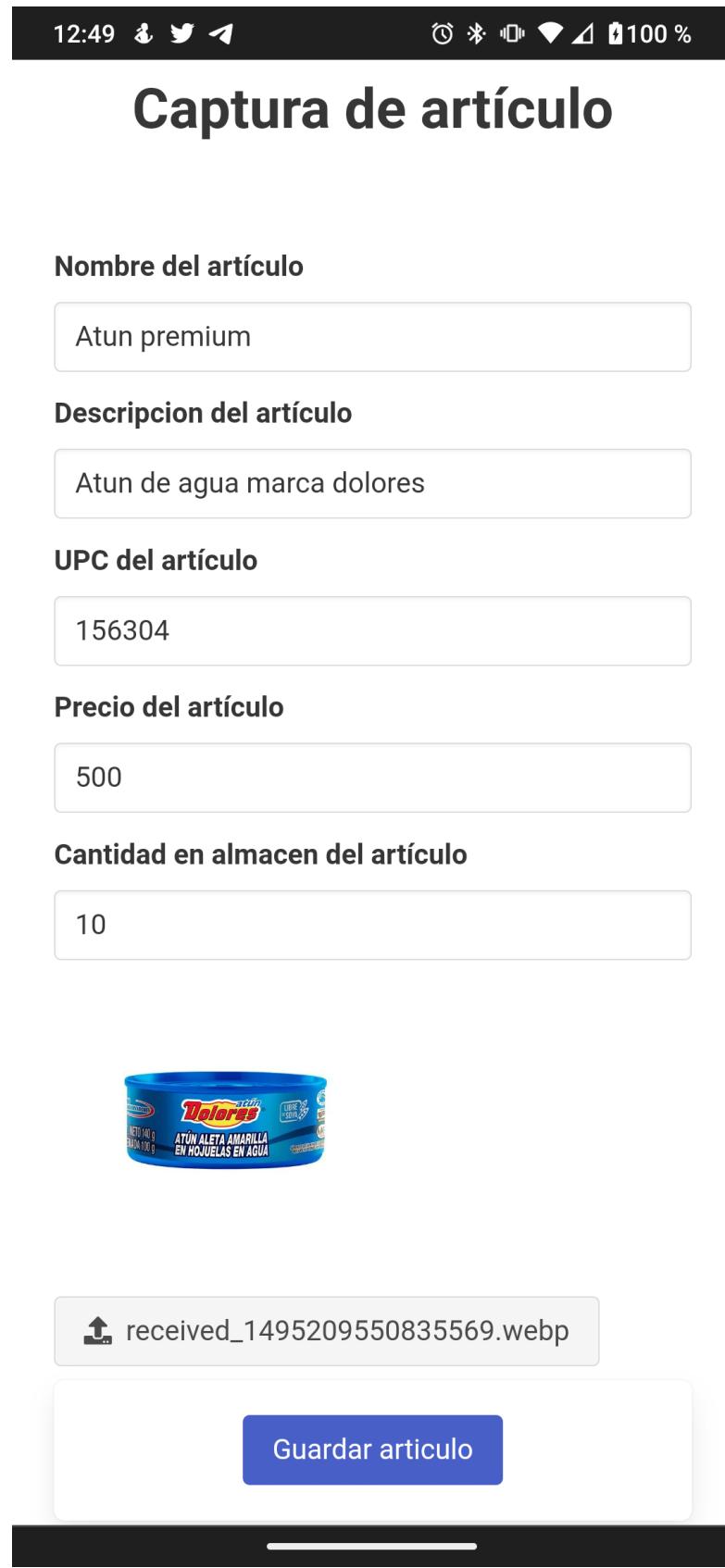


Figura 22: Prueba 1.4. Artículo dado de alta con foto en celular.

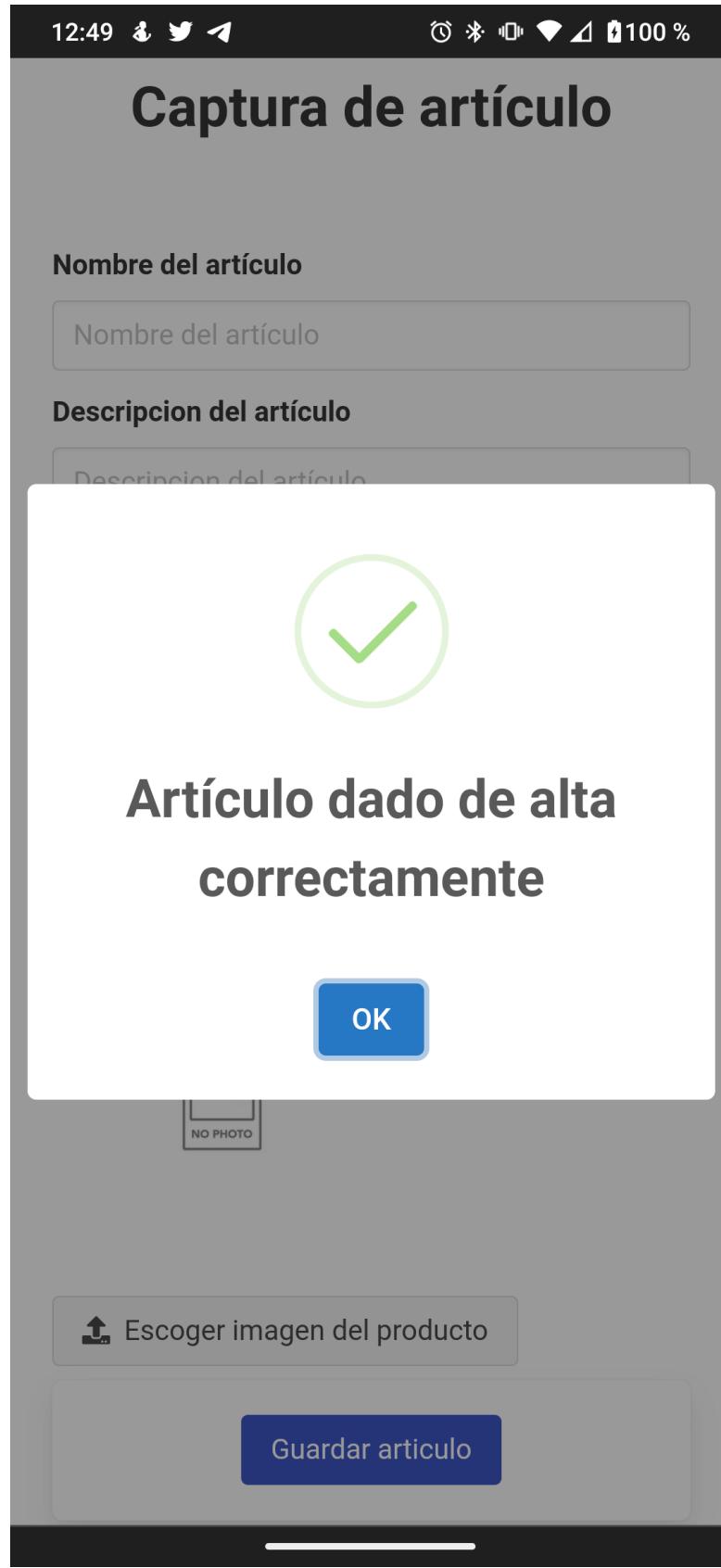


Figura 23: Resultado 1.4. Artículo dado de alta exitosamente con foto en celular.

Finalmente veamos la base de datos resultante de estas pruebas junto con otros artículos anteriormente creados, esto lo podemos ver en la figura 24 y en las figuras 25 a la 28 podemos ver la pagina de inicio y como podemos ver todos los artículos dados de alta, esto tanto en computadora como en celular.

```
edmundojm@COM2019630428 ~
mysql> select * from articulos;
+-----+-----+-----+-----+-----+
| id_articulo | nombre | descripcion | UPC | precio | cantidad_almacen |
+-----+-----+-----+-----+-----+
| 6 | Coca cola | coca cola de 3L retornable | 788204600 | 30.99 | 15400 |
| 7 | Coca cola | Coca cola light de 3L retornable | 9982033 | 30.25 | 12000 |
| 8 | Coca Cola | Coca Cola para taqueria de vidrio | 60066560 | 10.00 | 15000 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from articulos;
+-----+-----+-----+-----+-----+
| id_articulo | nombre | descripcion | UPC | precio | cantidad_almacen |
+-----+-----+-----+-----+-----+
| 6 | Coca cola | Coca cola de 3L retornable | 788204600 | 30.99 | 15400 |
| 7 | Coca cola | Coca Cola light de 3L retornable | 9982033 | 30.25 | 12000 |
| 8 | Coca Cola | Coca Cola para taqueria de vidrio | 60066560 | 10.00 | 15000 |
| 9 | Atun premium | Atun de aceite marca dolores | 66050500 | 500.00 | 12 |
| 10 | Atun premium | Atun de agua marca dolores | 156304 | 500.00 | 10 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Figura 24: Base de datos con los artículos dados de alta.

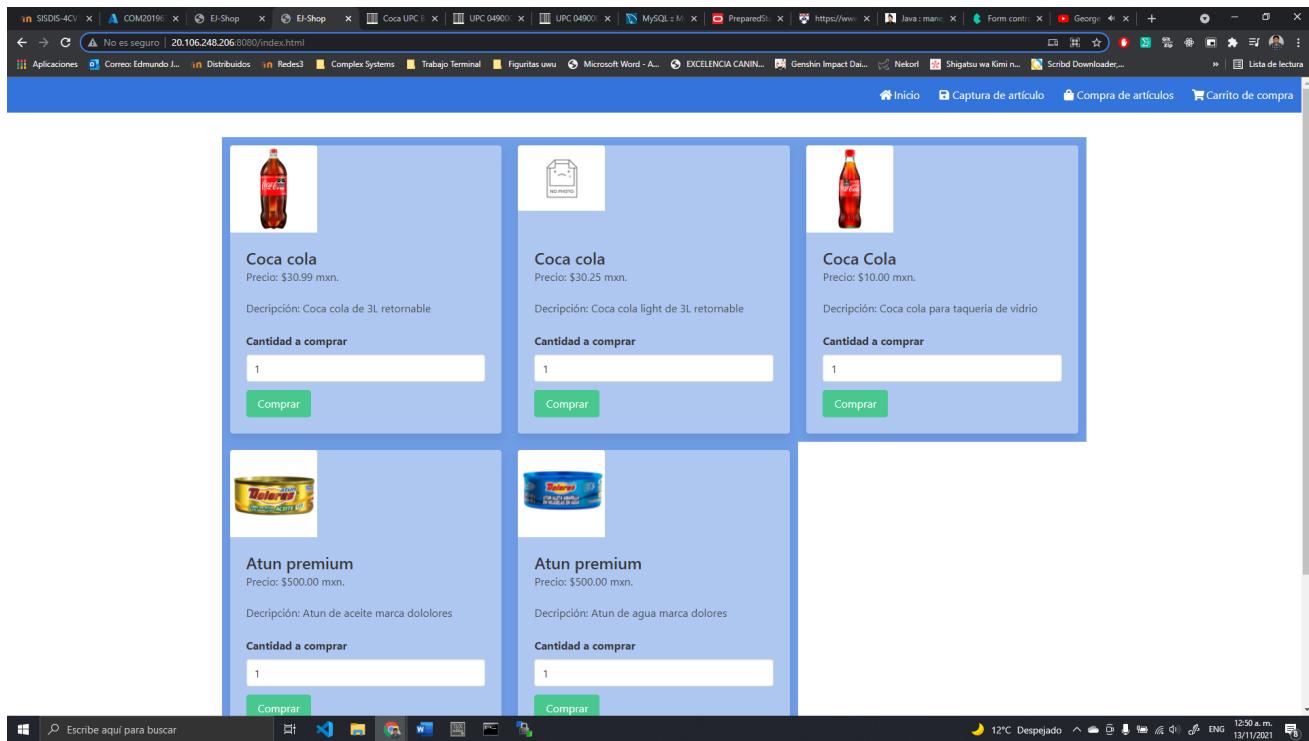


Figura 25: Pagina de inicio del sistema.

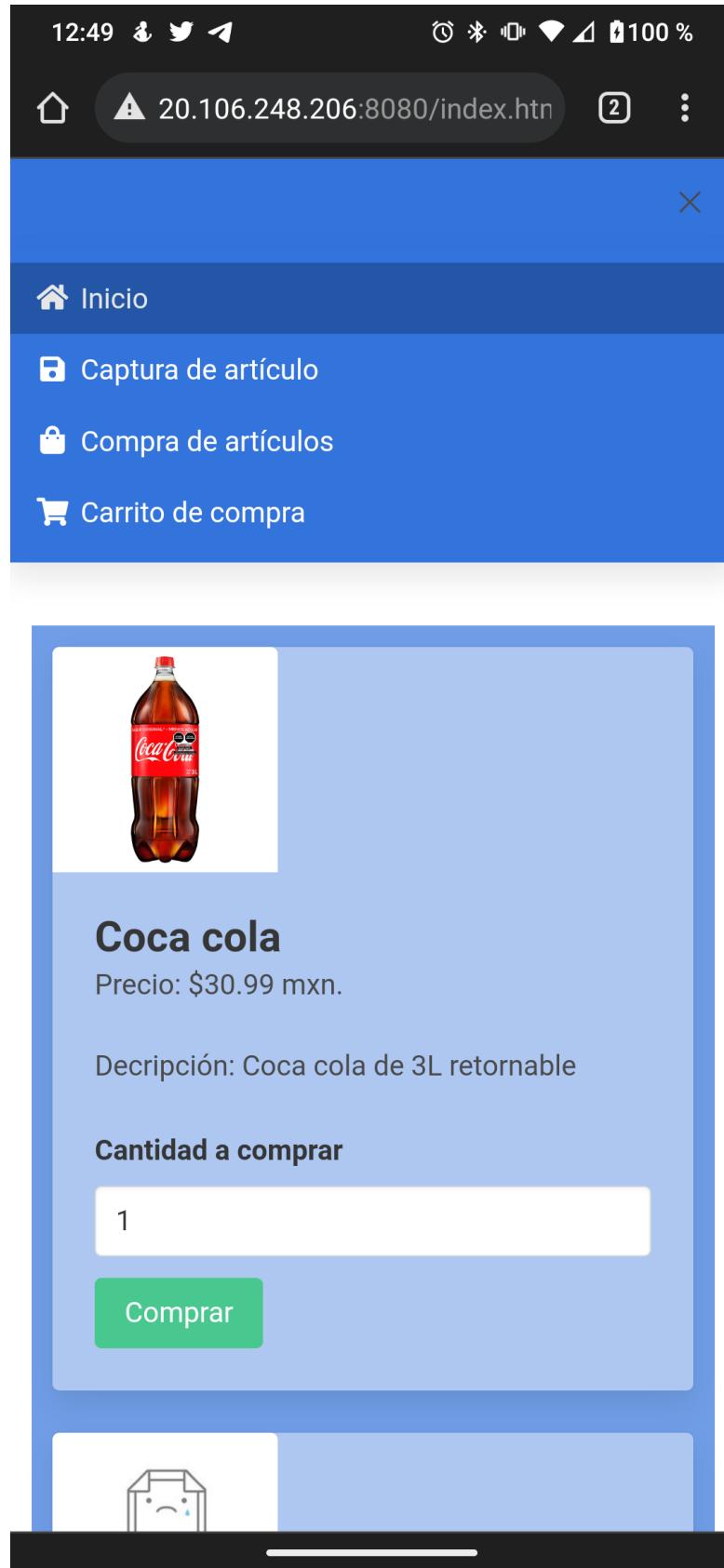


Figura 26: Pagina de inicio del sistema en celular. Parte 1.



Figura 27: Pagina de inicio del sistema en celular. Parte 2.



Figura 28: Pagina de inicio del sistema en celular. Parte 3.

3.2. Prueba 2. Búsqueda de artículos

Ahora vemos como se realizan la búsqueda de artículos, al dar click en la opción de Compra de artículos nos sale la opción de buscar artículos con base a una palabra ingresada, las pruebas que se hicieron fueron solo en la computadora, sin embargo, en la siguiente prueba que sera la compra de artículos se hacen también en el celular por lo cual cubrimos este punto. Ahora si veamos las pruebas, estas serán de la figura 29 a la 33.

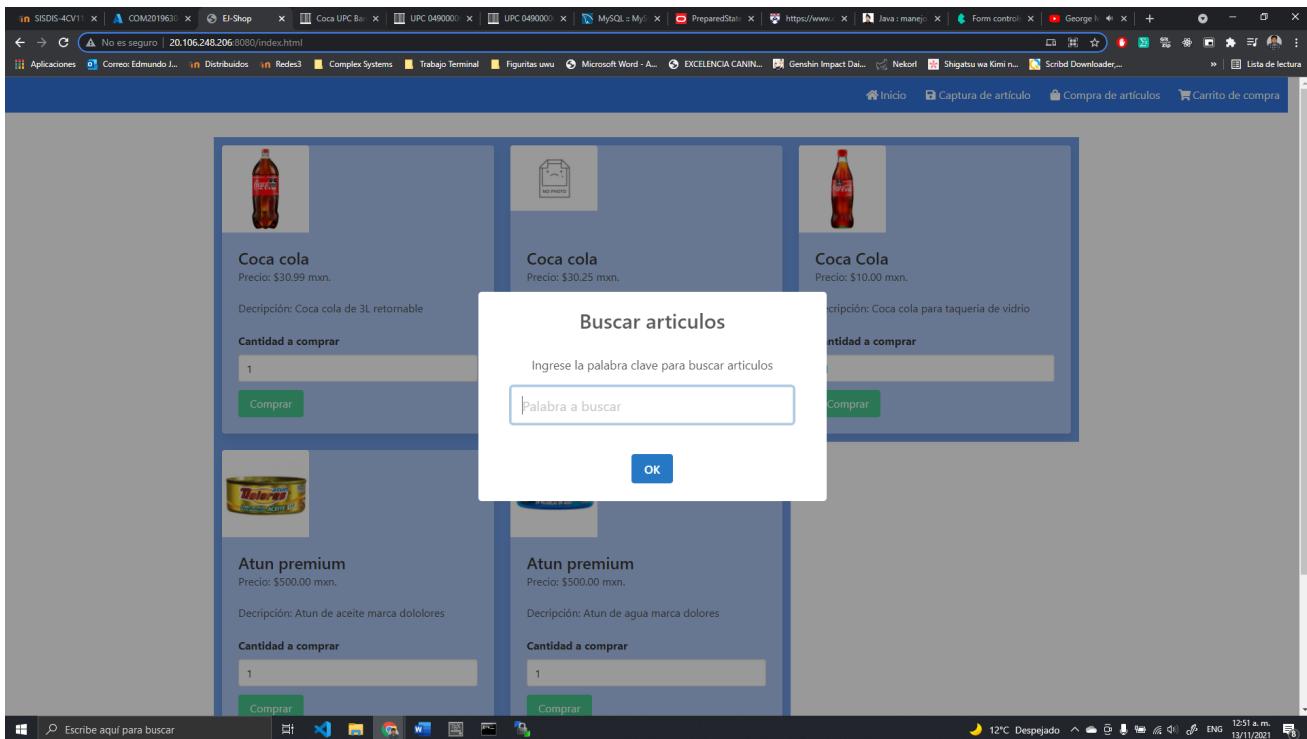


Figura 29: Ingresar palabra a buscar los artículos.

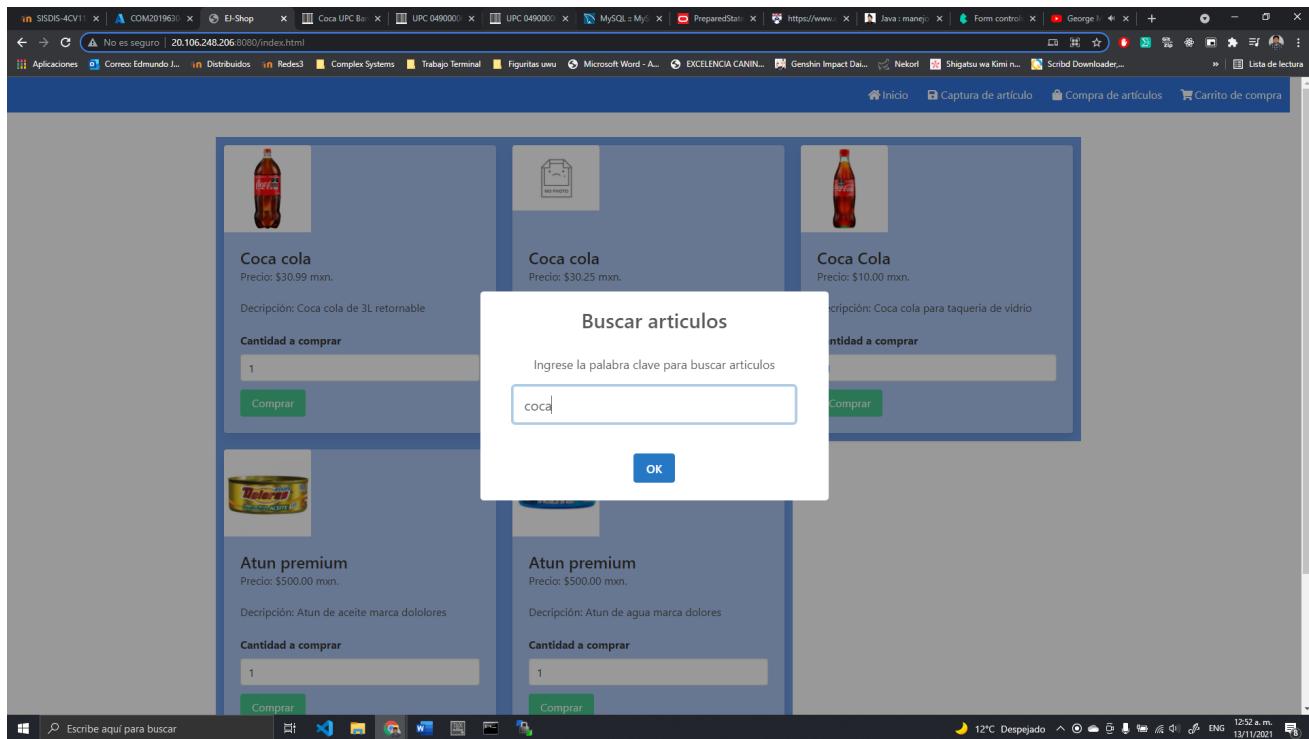


Figura 30: Buscando palabra “coca”.

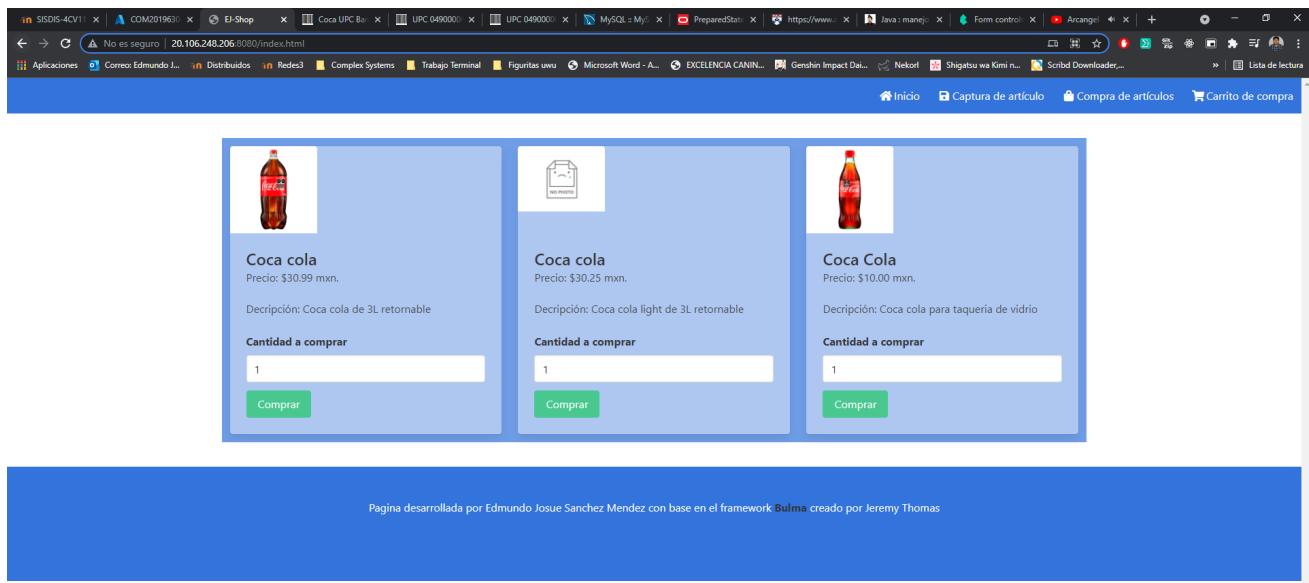


Figura 31: Resultado de la búsqueda de la palabra “coca”.

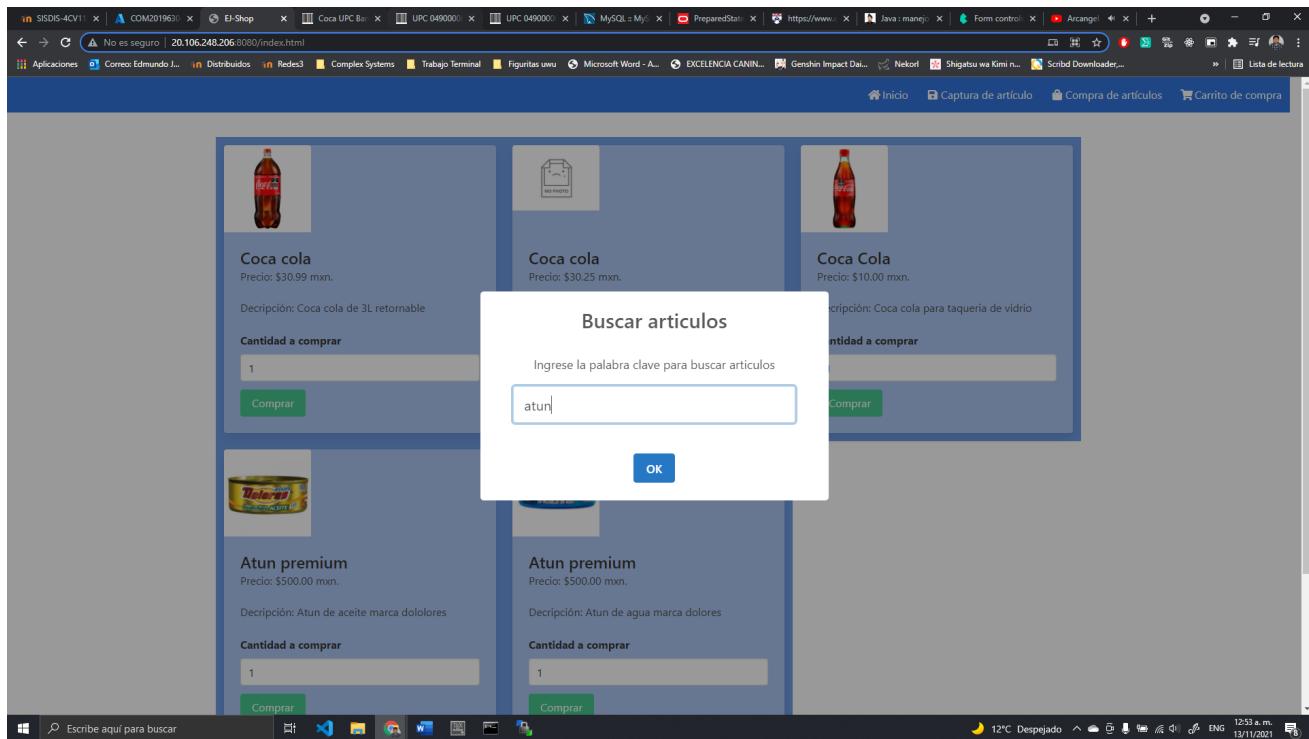


Figura 32: Buscando palabra “atun”.

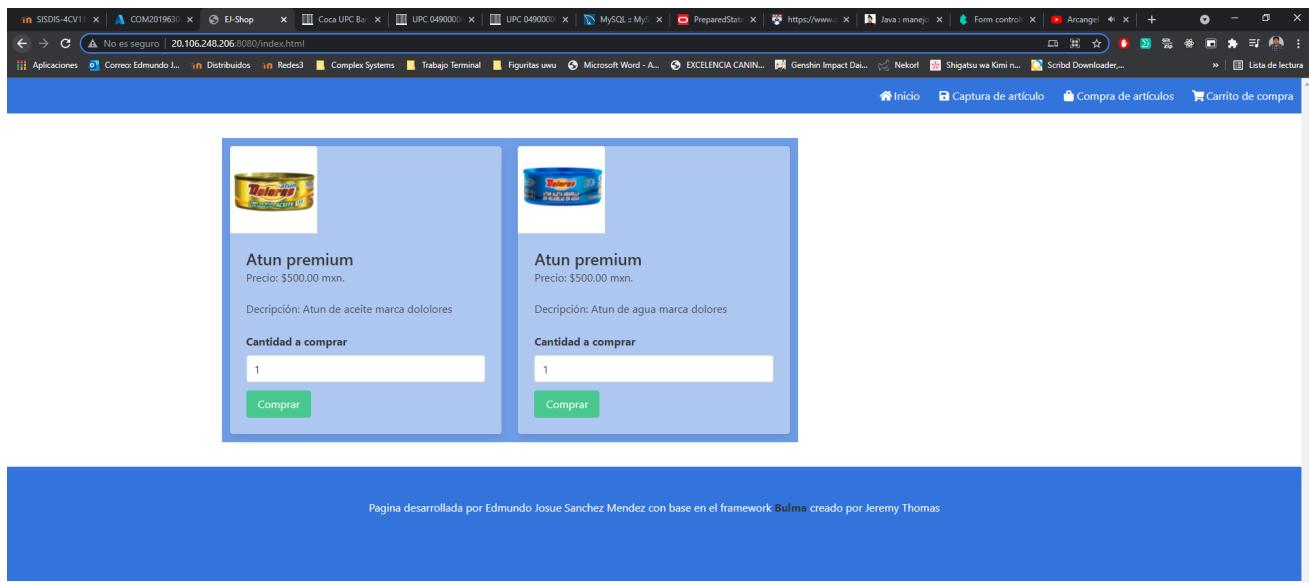


Figura 33: Resultado de la búsqueda de la palabra “atun”.

3.3. Prueba 3. Compra de artículos

Recordemos la figura 24 ya que nos sera útil.

La primera prueba sera intentar comprar una cantidad mayor de la que existe en la base de datos, esto en la figura 34 y vemos como en la figura 35 nos sale un error diciendo la cantidad disponible del articulo.

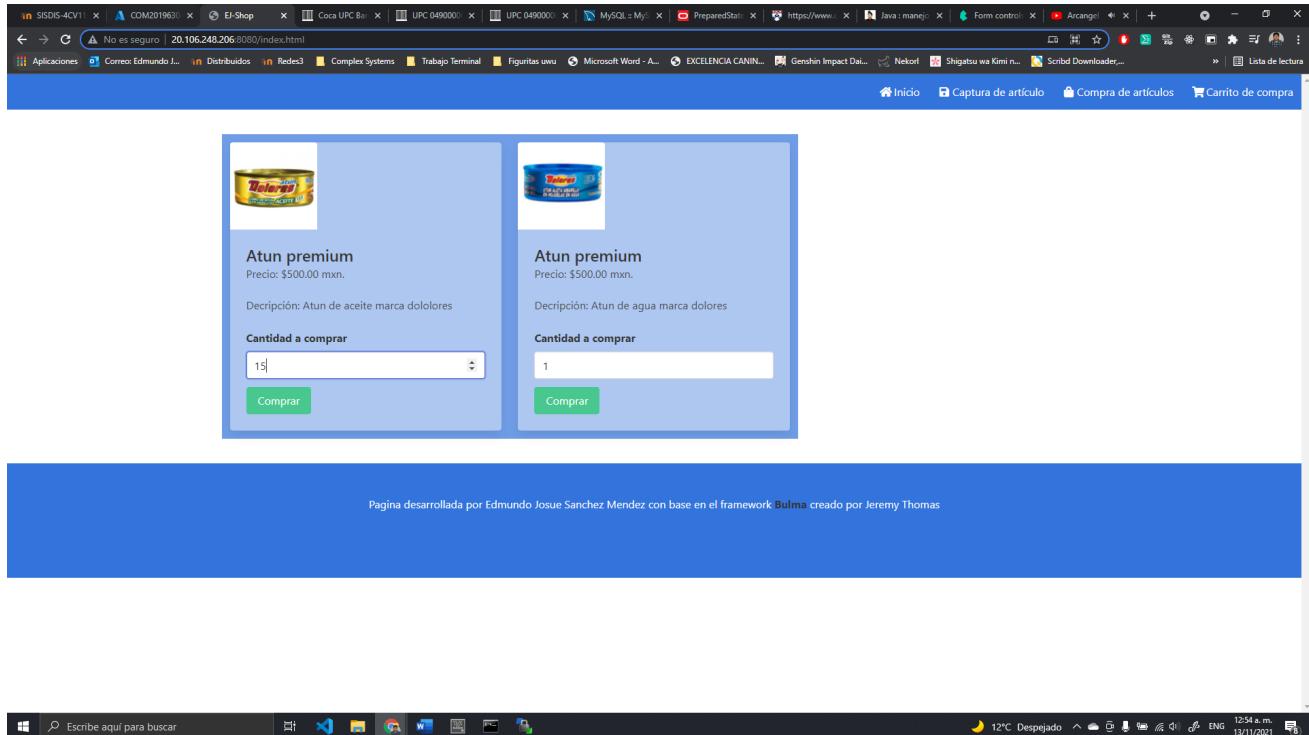


Figura 34: Intento de comprar una cantidad mayor de la existente.

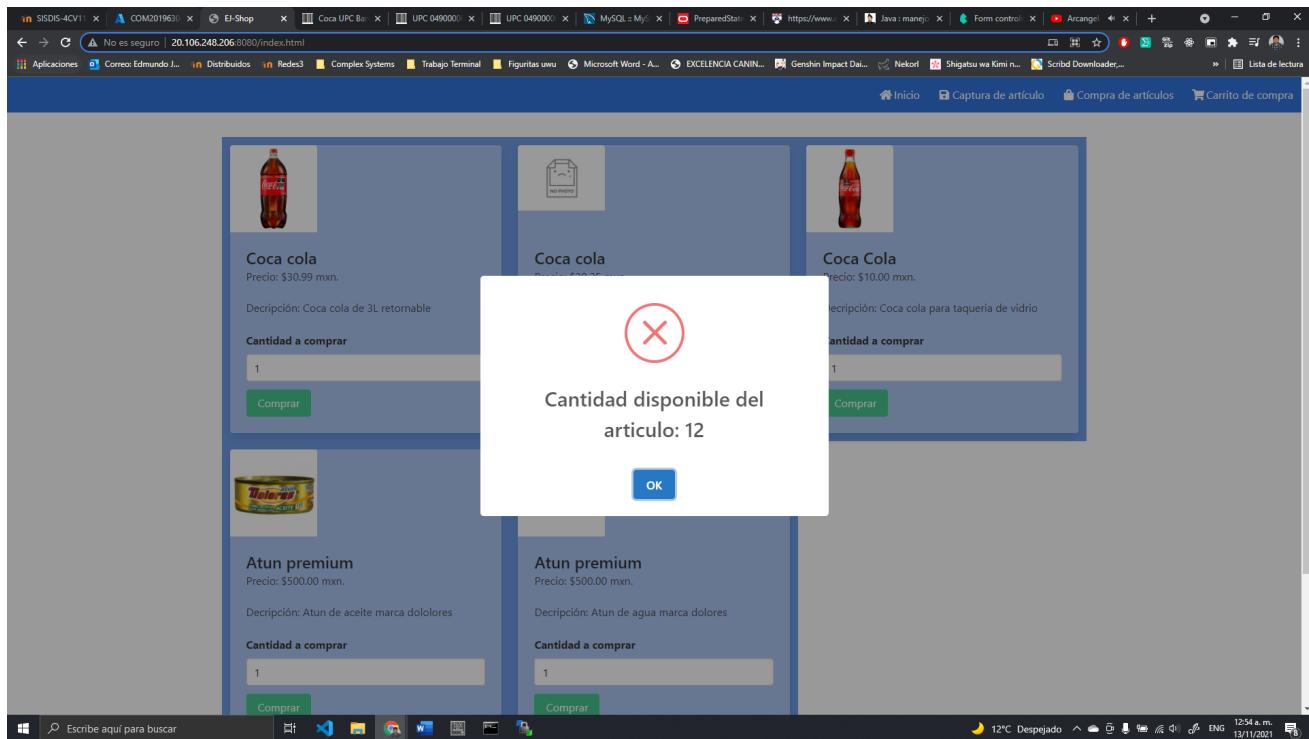


Figura 35: Error al comprar, ya que no hay suficiente cantidad para cubrir la compra.

En las siguientes dos figuras vemos como podemos comprar si la cantidad solicitada es menor o igual a la existente

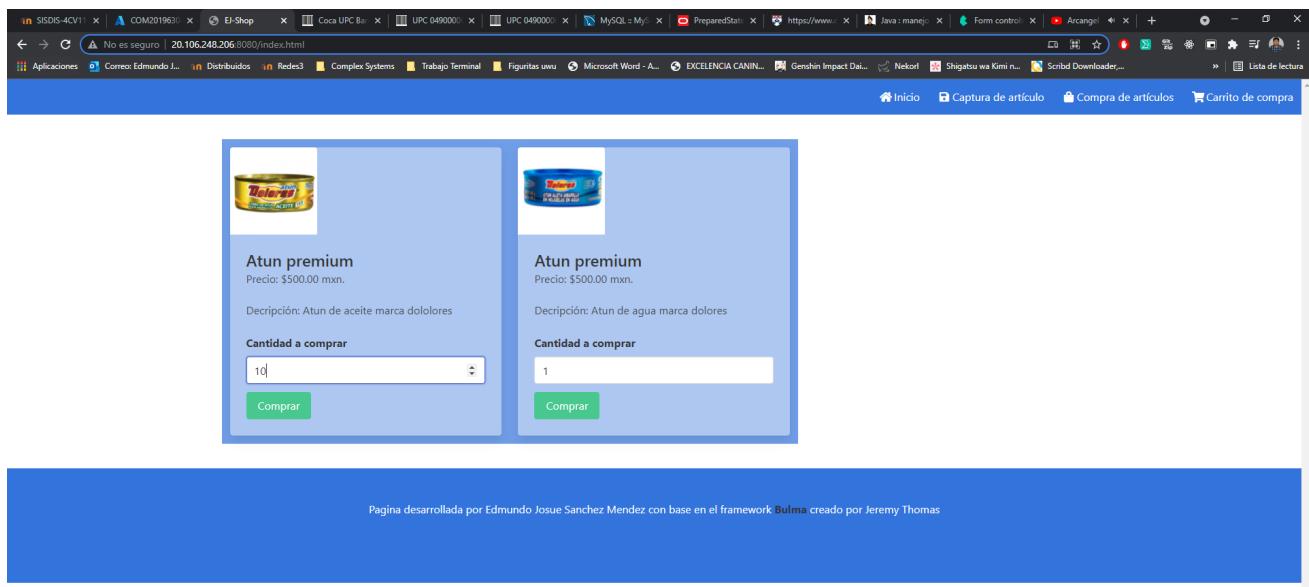


Figura 36: Cantidad a comprar de 10.

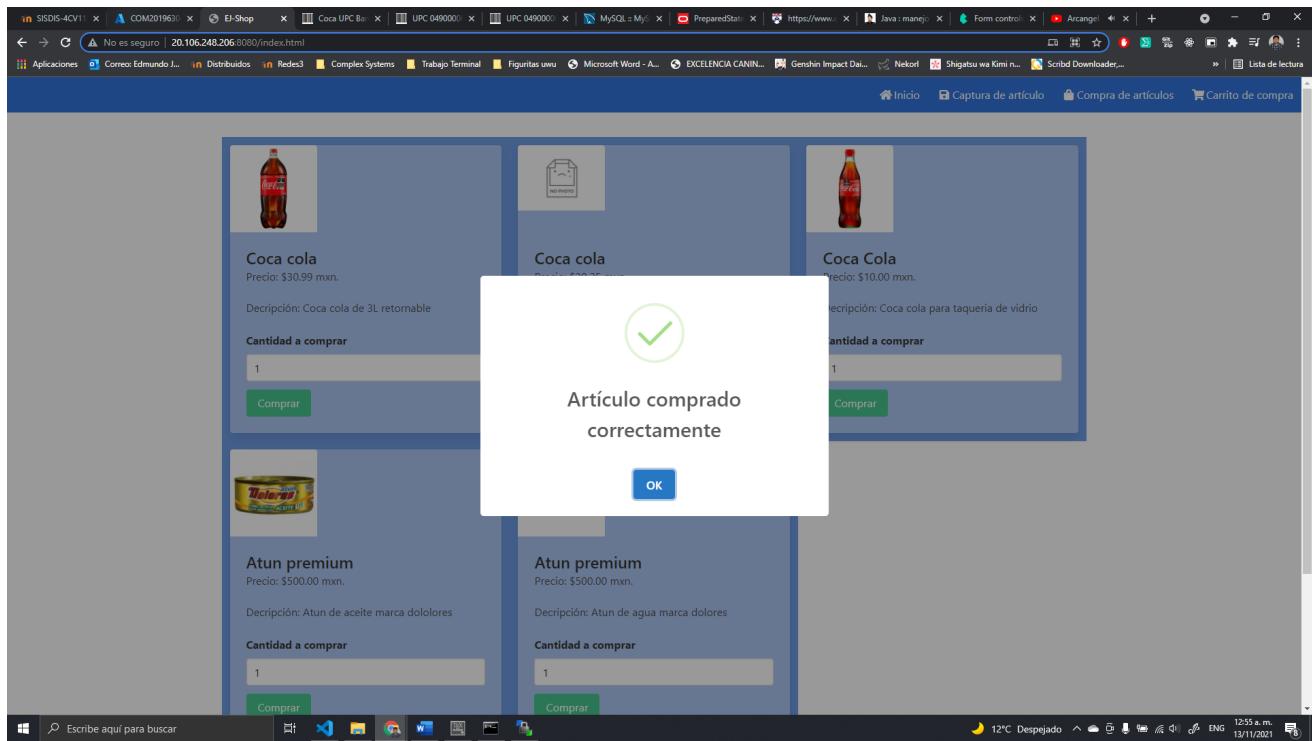


Figura 37: Compra realizada con éxito.

Una prueba mas es la que podemos ver en las siguientes 2 figuras comprando directamente de la pagina de inicio, esto para poder ver que la funcionalidad de compra también esta disponible en esta opción, sin embargo, hay un pequeño detalle y es que podemos comprar varias veces un mismo articulo siempre y cuando podamos cubrir la cantidad solicitada, es por eso que se realiza esta prueba.

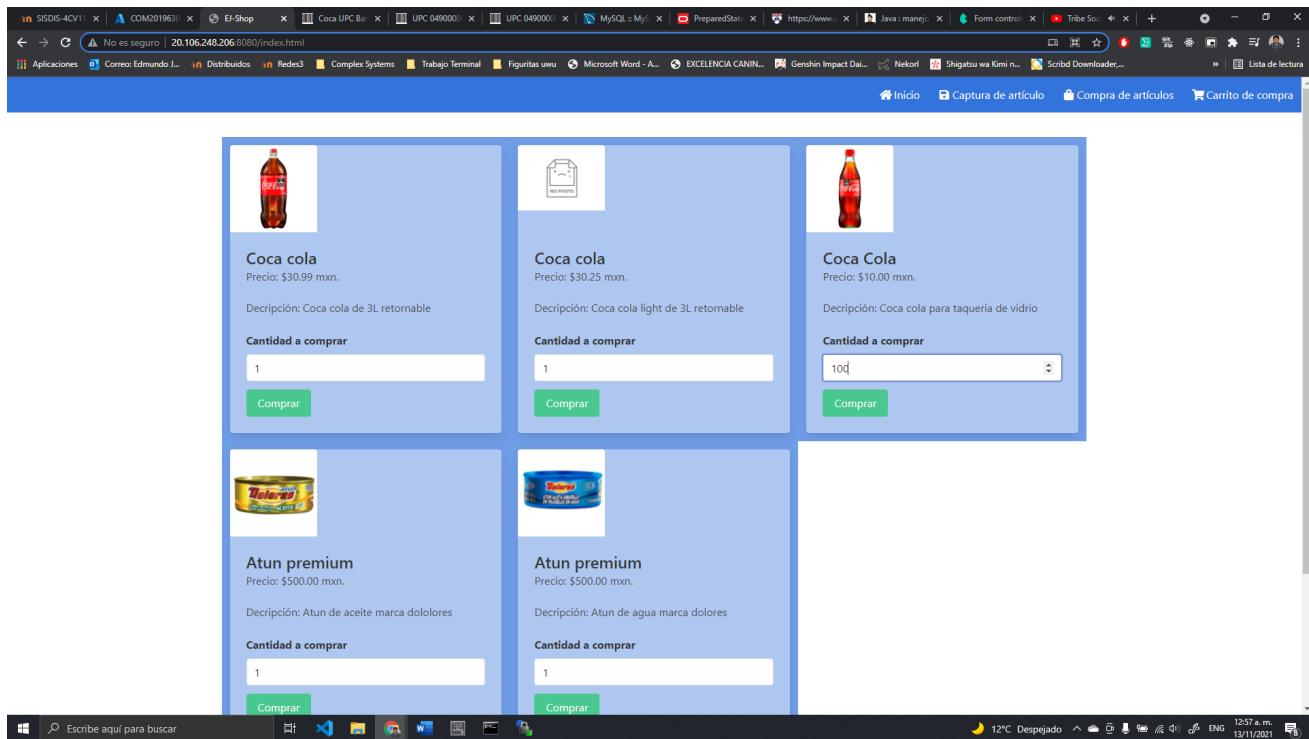


Figura 38: Re compra de un articulo.

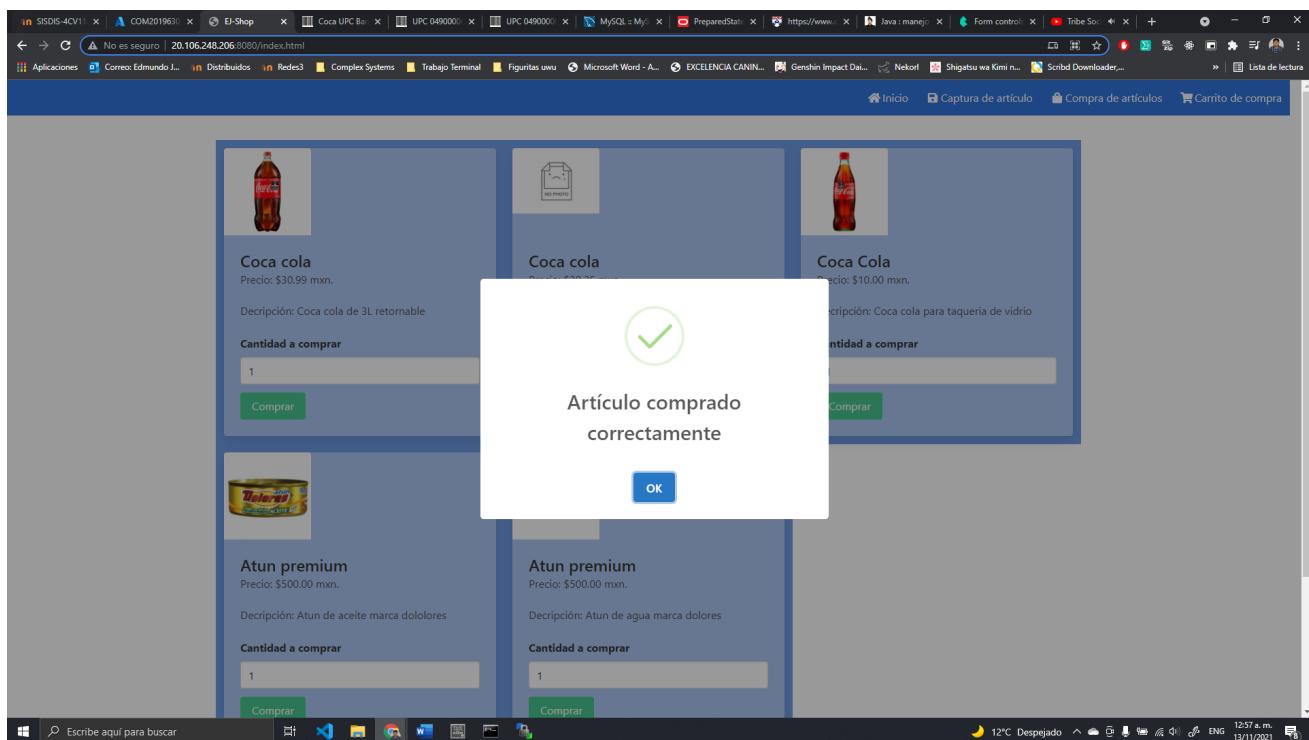


Figura 39: Re compra de un articulo realizada con éxito.

Y como podemos ver en la figura 40 la base de datos se fue actualizando con base en las compras realizadas, tanto en la tabla de artículos como en la tabla de carrito_compra.

```
edmundojm@COM2019639428:~ %
5 | Coca cola | Coca cola de 3L retornable | 788204600 | 30.99 | 15400 |
6 | Coca cola | Coca cola light de 3L retornable | 9982033 | 30.25 | 12000 |
7 | Coca Cola | Coca cola para taqueria de vidrio | 60066560 | 10.00 | 15000 |
8 | Atun premium | Atun de aceite marca dolores | 66050500 | 500.00 | 2 |
9 | Atun premium | Atun de agua marca dolores | 156304 | 500.00 | 10 |
10 | Atun premium | Atun de agua marca dolores | 156304 | 500.00 | 10 |
5 rows in set (0.00 sec)

mysql> select * from articulos;
+----+-----+-----+-----+-----+
| id_articulo | nombre | descripcion | UPC | precio | cantidad_almacen |
+----+-----+-----+-----+-----+
| 6 | Coca cola | Coca cola de 3L retornable | 788204600 | 30.99 | 15400 |
| 7 | Coca cola | Coca cola light de 3L retornable | 9982033 | 30.25 | 12000 |
| 8 | Coca Cola | Coca cola para taqueria de vidrio | 60066560 | 10.00 | 15000 |
| 9 | Atun premium | Atun de aceite marca dolores | 66050500 | 500.00 | 2 |
| 10 | Atun premium | Atun de agua marca dolores | 156304 | 500.00 | 10 |
5 rows in set (0.00 sec)

mysql> select * from carrito_compra;
+----+-----+
| id_compra | cantidad | id_articulo |
+----+-----+
| 1 | 10 | 9 |
1 row in set (0.00 sec)

mysql> select * from articulos;
+----+-----+-----+-----+-----+
| id_articulo | nombre | descripcion | UPC | precio | cantidad_almacen |
+----+-----+-----+-----+-----+
| 6 | Coca cola | Coca cola de 3L retornable | 788204600 | 30.99 | 15395 |
| 7 | Coca cola | Coca cola light de 3L retornable | 9982033 | 30.25 | 12000 |
| 8 | Coca Cola | Coca cola para taqueria de vidrio | 60066560 | 10.00 | 14800 |
| 9 | Atun premium | Atun de aceite marca dolores | 66050500 | 500.00 | 2 |
| 10 | Atun premium | Atun de agua marca dolores | 156304 | 500.00 | 10 |
5 rows in set (0.00 sec)

mysql> select * from carrito_compra;
+----+-----+
| id_compra | cantidad | id_articulo |
+----+-----+
| 1 | 10 | 9 |
| 2 | 5 | 6 |
| 3 | 100 | 8 |
| 4 | 100 | 8 |
+----+-----+
4 rows in set (0.00 sec)

Windows Escribe aquí para buscar 12°C Despejado 12:38 a.m. ENG 13/11/2021
```

Figura 40: Re compra de un artículo realizada con éxito.

Ahora en las siguientes tres imágenes veremos la prueba de compra desde el celular junto con la opción de búsqueda de artículo.

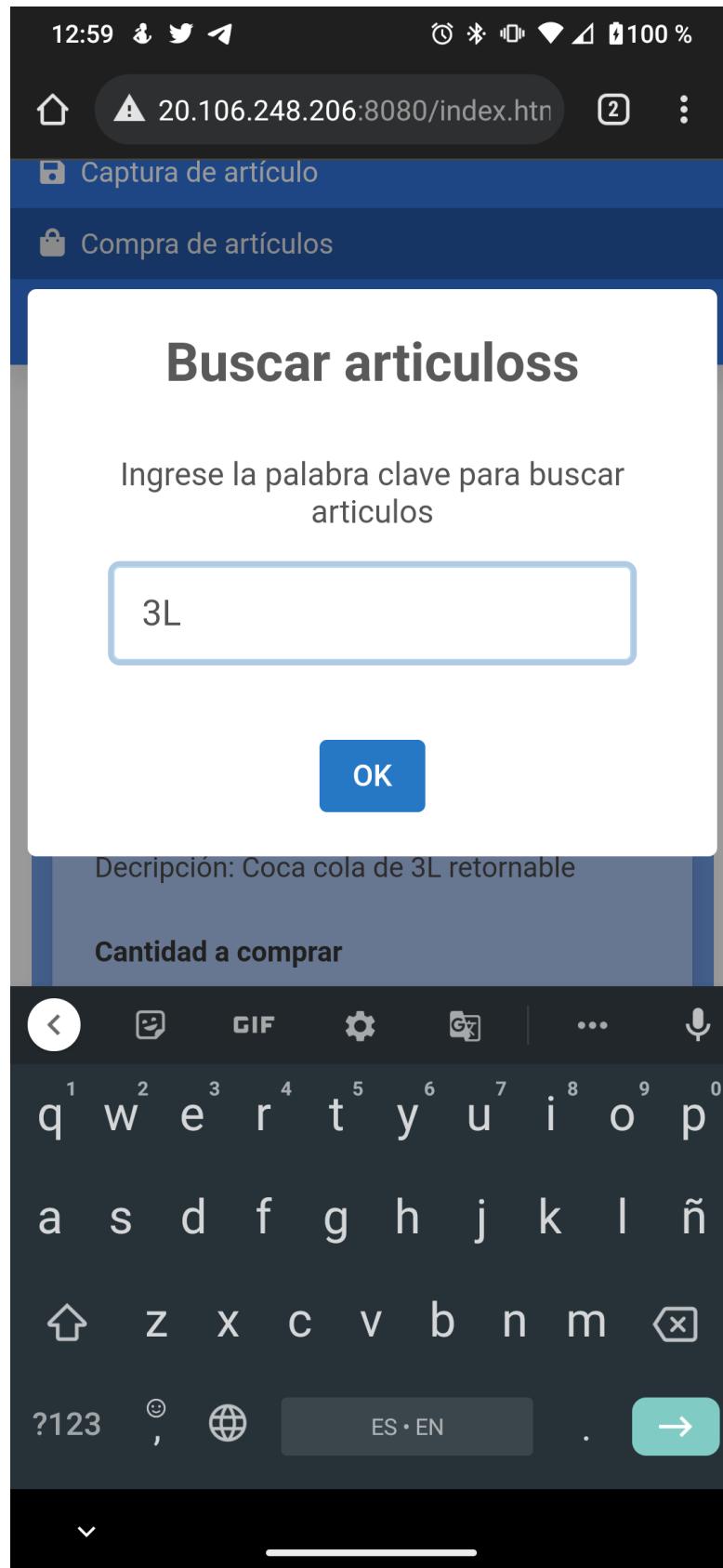


Figura 41: Busqueda de artículos con la palabra 3L.



Figura 42: Desplegado de los artículos.

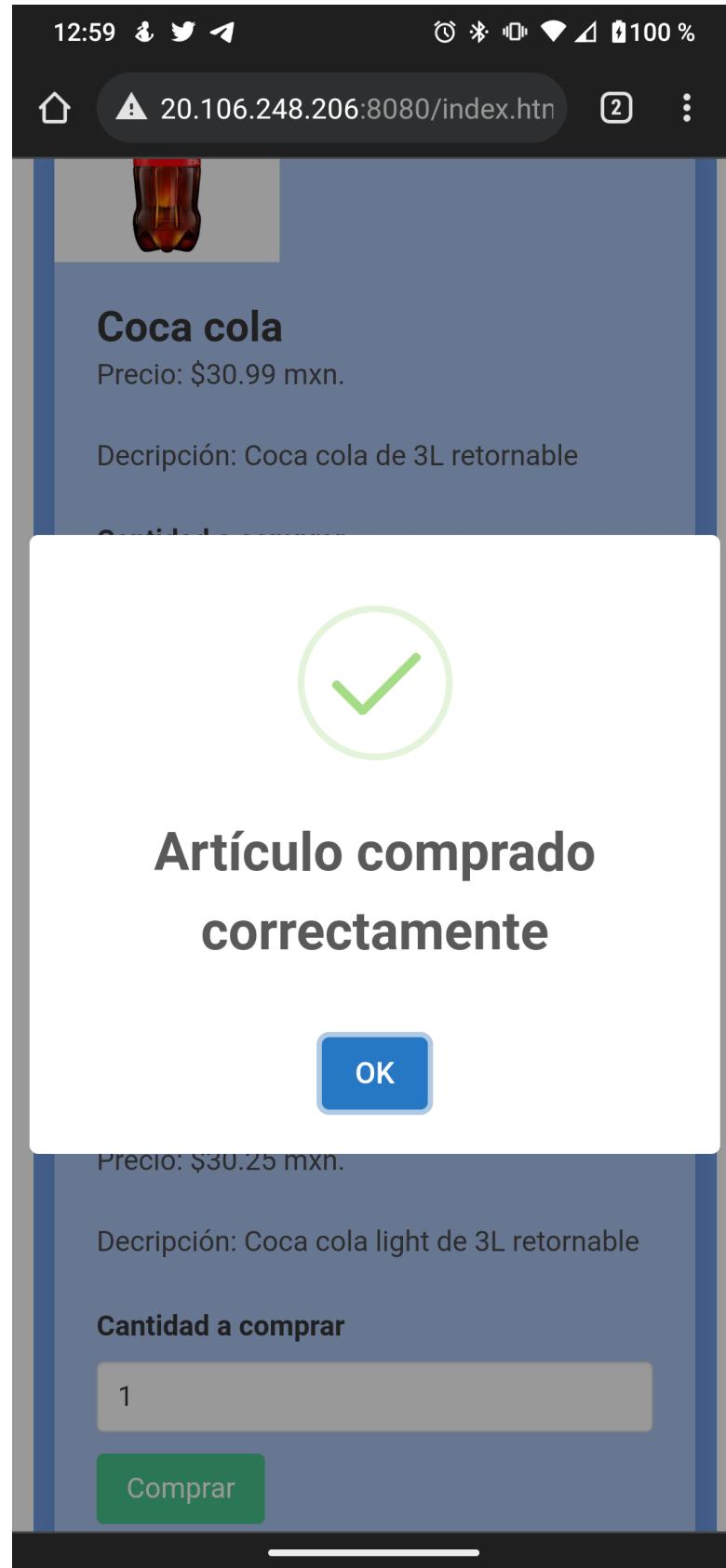
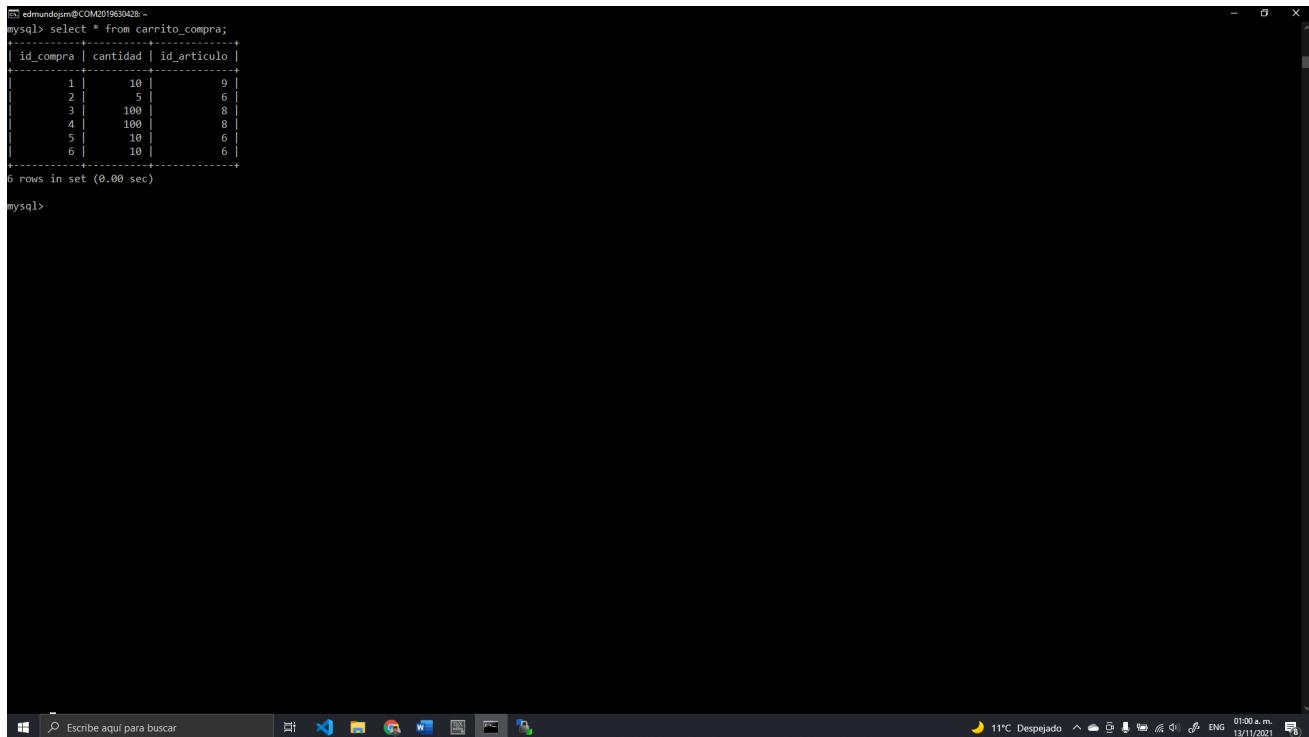


Figura 43: Compra realizada con exitoso.

Y como podemos ver en la figura 44 la base de datos se fue actualizando con base en al compras realizadas, tanto en la tabla de artículos como en la tabla de carrito_compra.



```
edmundojm@COM2019630428:~  
mysql> select * from carrito_compra;  
+----+-----+-----+  
| id_compra | cantidad | id_articulo |  
+----+-----+-----+  
| 1 | 10 | 9 |  
| 2 | 5 | 6 |  
| 3 | 100 | 8 |  
| 4 | 100 | 8 |  
| 5 | 10 | 6 |  
| 6 | 10 | 6 |  
+----+-----+-----+  
6 rows in set (0.00 sec)  
mysql>
```

Figura 44: Base de datos actualizada.

Ahora en las siguientes 3 figuras veamos una vez el intento de compra de mas artículos de los disponibles, como vemos en la base de datos no hay cambio alguno.

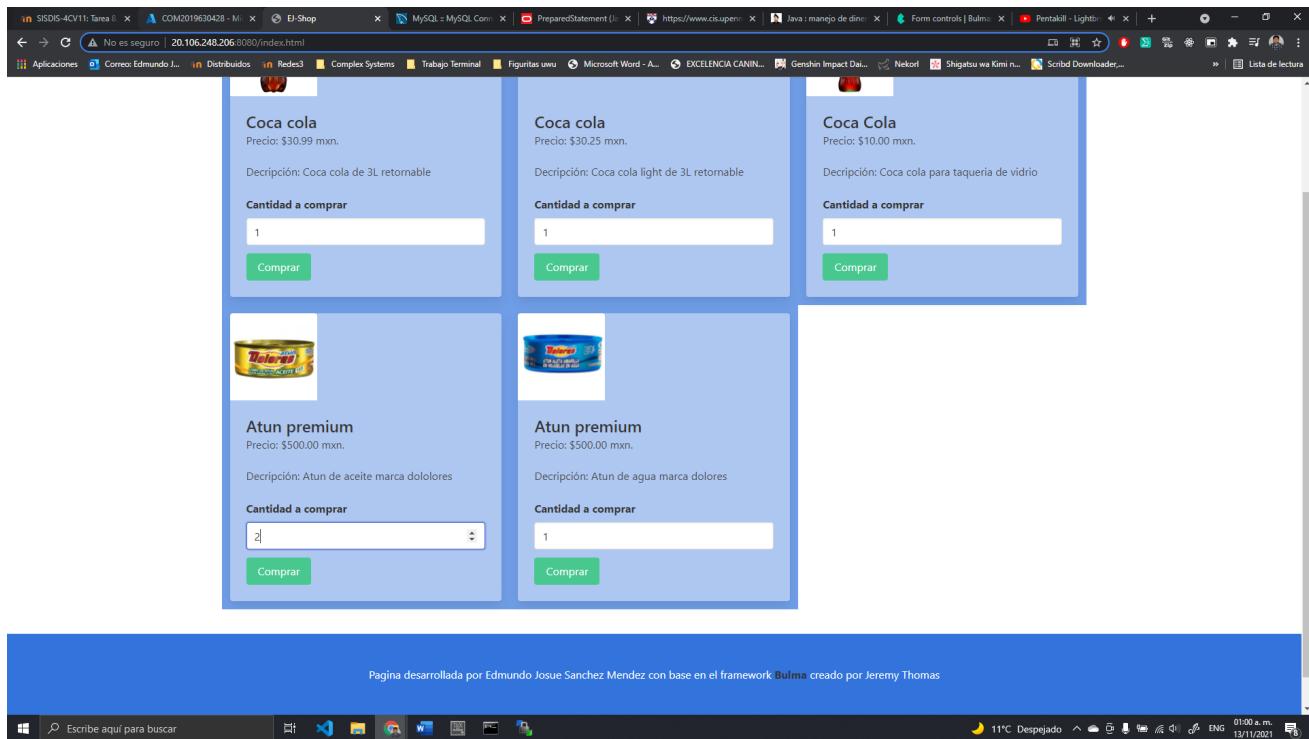


Figura 45: Intento de compra de mas producto.

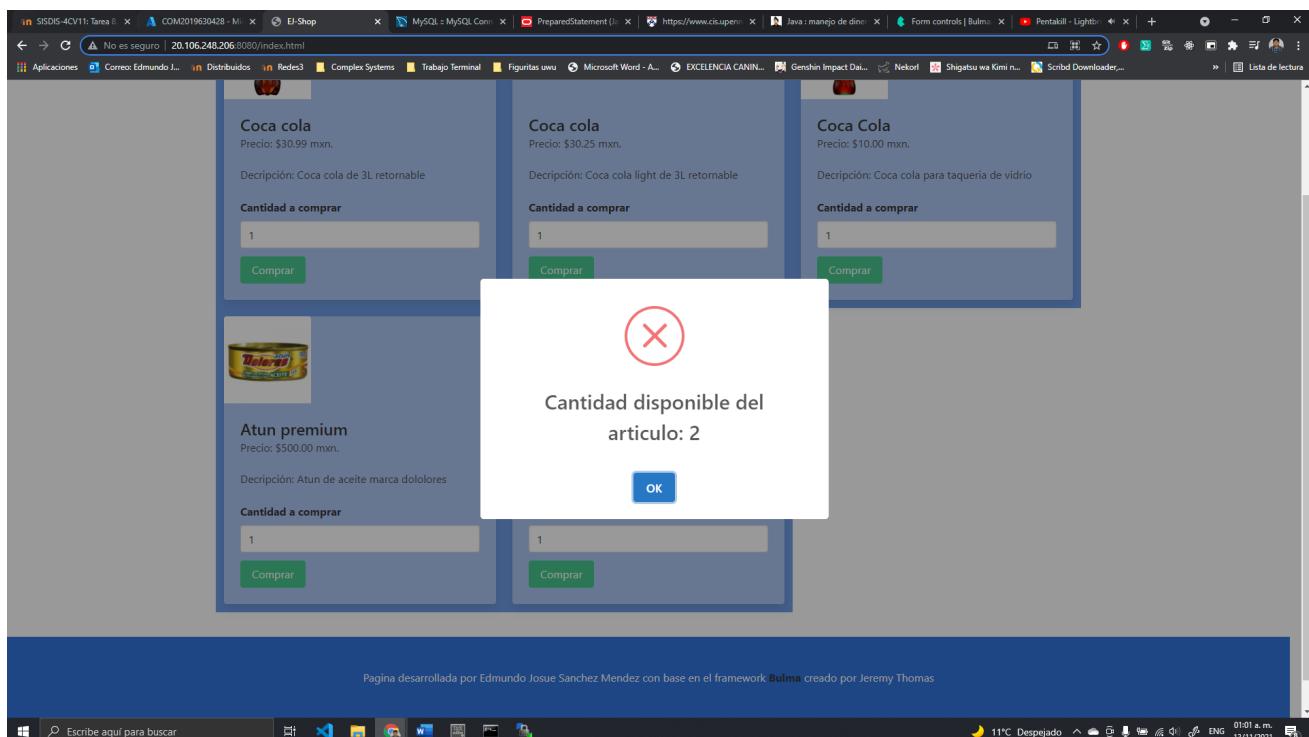


Figura 46: Respuesta del sistema.

```

edmundojm@COM2019630428:~ mysql> select * from carrito_compra;
+-----+-----+-----+
| id_compra | cantidad | id_articulo |
+-----+-----+-----+
| 1 | 10 | 9 |
| 2 | 5 | 6 |
| 3 | 100 | 8 |
| 4 | 100 | 8 |
| 5 | 10 | 6 |
| 6 | 10 | 6 |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from articulos;
+-----+-----+-----+-----+-----+
| id_articulo | nombre | descripcion | UPC | precio | cantidad_almacen |
+-----+-----+-----+-----+-----+
| 6 | Coca cola | Coca cola de 3L retornable | 788204600 | 30.99 | 15375 |
| 7 | Coca cola | Coca cola light de 3L retornable | 9982033 | 30.25 | 12000 |
| 8 | Coca Cola | Coca cola para taqueria de vidrio | 60066560 | 10.00 | 14800 |
| 9 | Atun premium | Atun de aceite marca dolores | 66050500 | 500.00 | 2 |
| 10 | Atun premium | Atun de agua marca dolores | 156304 | 500.00 | 10 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from articulos;
+-----+-----+-----+-----+-----+
| id_articulo | nombre | descripcion | UPC | precio | cantidad_almacen |
+-----+-----+-----+-----+-----+
| 6 | Coca cola | Coca cola de 3L retornable | 788204600 | 30.99 | 15375 |
| 7 | Coca cola | Coca cola light de 3L retornable | 9982033 | 30.25 | 12000 |
| 8 | Coca Cola | Coca cola para taqueria de vidrio | 60066560 | 10.00 | 14800 |
| 9 | Atun premium | Atun de aceite marca dolores | 66050500 | 500.00 | 2 |
| 10 | Atun premium | Atun de agua marca dolores | 156304 | 500.00 | 10 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>

```

Figura 47: Base de datos no modificada.

Finalmente en las siguientes 3 figuras veamos una vez el intento de compra de la cantidad artículos exacta que están disponibles, como vemos en la base de datos no hay cambio alguno.

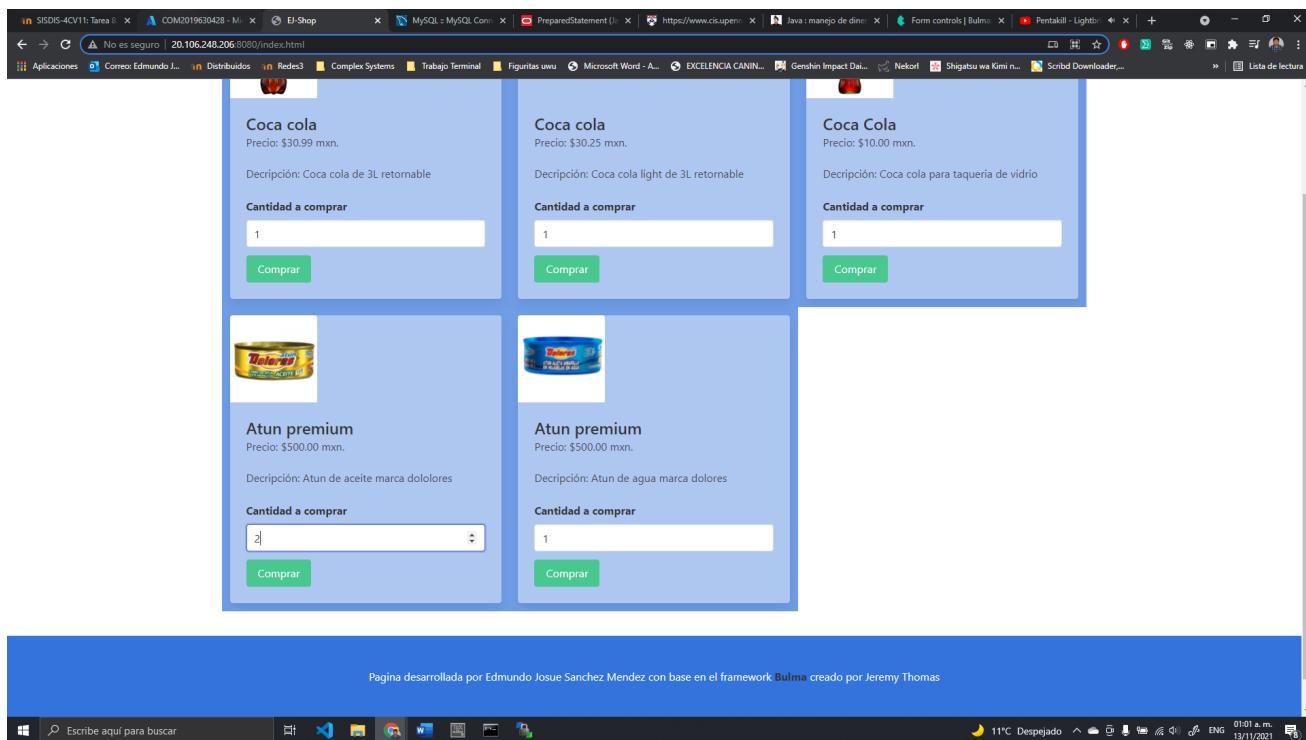


Figura 48: Intento de compra de producto exacto.

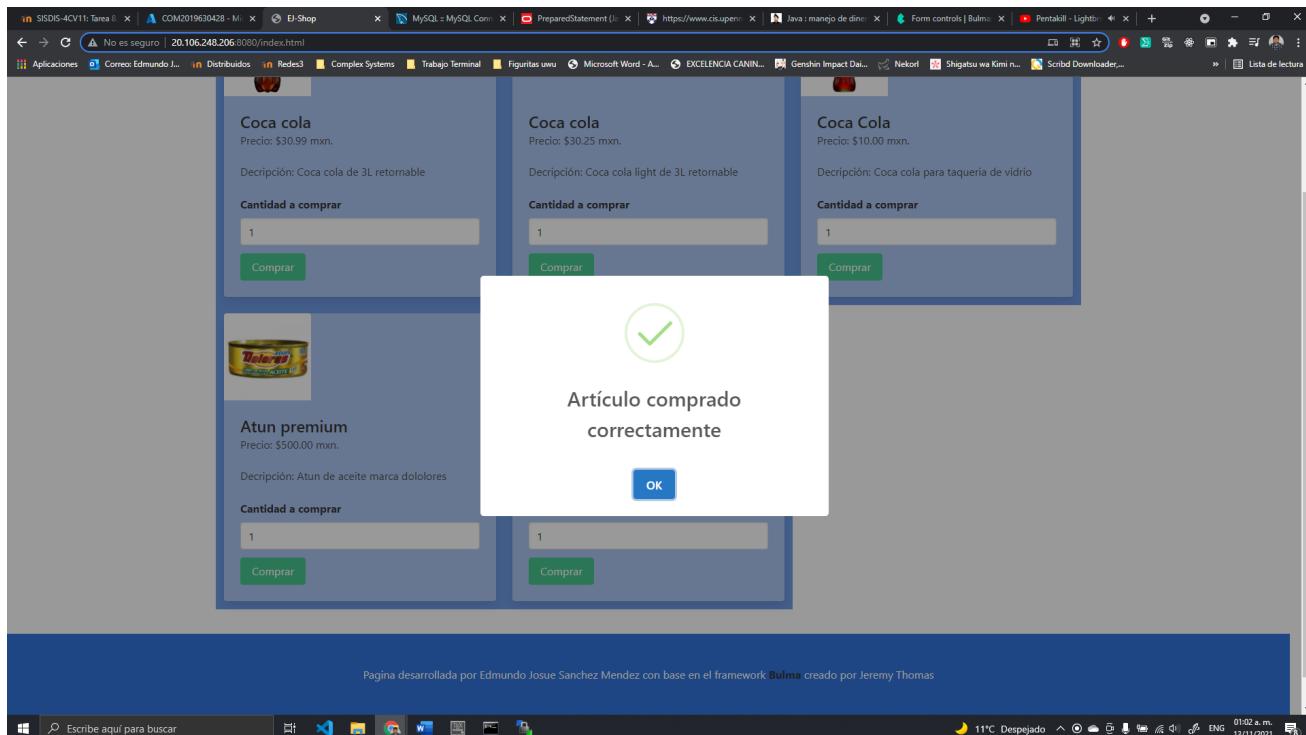


Figura 49: Respuesta del sistema.

```

edmundojm@COM2019E0428:~ | 6 | 10 | 6 |
6 rows in set (0.00 sec)

mysql> select * from articulos;
+-----+-----+-----+-----+-----+
| id_articulo | nombre | descripcion | UPC | precio | cantidadalmacen |
+-----+-----+-----+-----+-----+
| 6 | Coca cola | Coca cola de 3L retornable | 788204600 | 30.99 | 15375 |
| 7 | Coca cola | Coca cola light de 3L retornable | 9982033 | 30.25 | 12000 |
| 8 | Coca Cola | Coca Cola para taqueria de vidrio | 60066560 | 10.00 | 14800 |
| 9 | Atun premium | Atun de aceite marca dolores | 66050500 | 500.00 | 2 |
| 10 | Atun premium | Atun de agua marca dolores | 156304 | 500.00 | 10 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from articulos;
+-----+-----+-----+-----+-----+
| id_articulo | nombre | descripcion | UPC | precio | cantidadalmacen |
+-----+-----+-----+-----+-----+
| 6 | Coca cola | Coca cola de 3L retornable | 788204600 | 30.99 | 15375 |
| 7 | Coca cola | Coca cola light de 3L retornable | 9982033 | 30.25 | 12000 |
| 8 | Coca Cola | Coca Cola para taqueria de vidrio | 60066560 | 10.00 | 14800 |
| 9 | Atun premium | Atun de aceite marca dolores | 66050500 | 500.00 | 2 |
| 10 | Atun premium | Atun de agua marca dolores | 156304 | 500.00 | 10 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from articulos;
+-----+-----+-----+-----+-----+
| id_articulo | nombre | descripcion | UPC | precio | cantidadalmacen |
+-----+-----+-----+-----+-----+
| 6 | Coca cola | Coca cola de 3L retornable | 788204600 | 30.99 | 15375 |
| 7 | Coca cola | Coca cola light de 3L retornable | 9982033 | 30.25 | 12000 |
| 8 | Coca Cola | Coca Cola para taqueria de vidrio | 60066560 | 10.00 | 14800 |
| 9 | Atun premium | Atun de aceite marca dolores | 66050500 | 500.00 | 0 |
| 10 | Atun premium | Atun de agua marca dolores | 156304 | 500.00 | 10 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>

```

Figura 50: Base de datos modificada y con 0 en cantidad restante.

3.4. Prueba 4. Carrito de compra

Ahora veamos como se ve el carrito de compra, en las figuras siguientes veremos como se ve el carrito de compras tanto en celular como en celular.

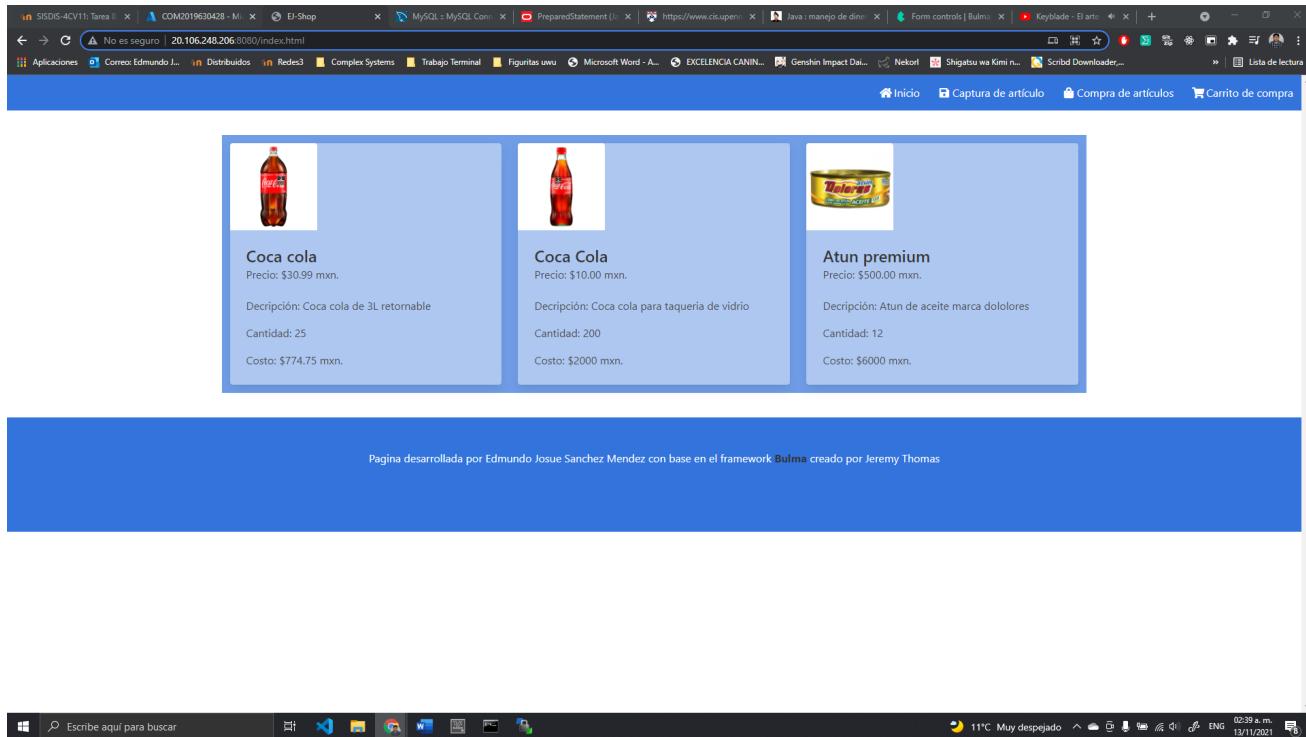


Figura 51: Desplegado del carrito de compra en computadora.

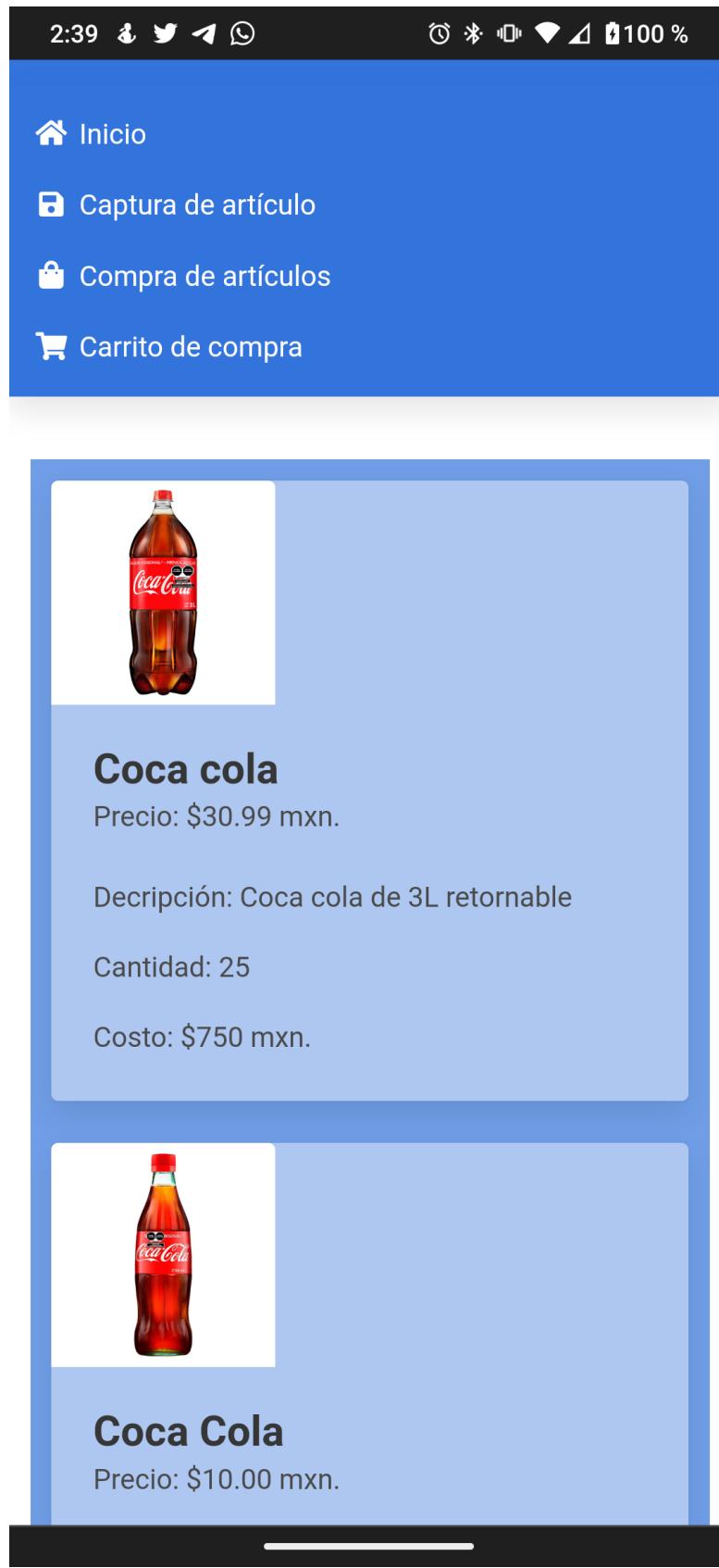


Figura 52: Desplegado del carrito de compra en celular.



Figura 53: Desplegado del carrito de compra en celular.

Finalmente veamos la base de datos resultante del carrito de compra.

```

edmundospn@CCM2019630428:~/Servicio
mysql> select * from articulos;
+----+-----+-----+-----+-----+-----+
| id_articulo | nombre | descripcion | UPC | precio | cantidad_almacen |
+----+-----+-----+-----+-----+-----+
| 6 | Coca cola | Coca cola de 3l retornable | 788204600 | 30.99 | 15375 |
| 7 | Coca cola | Coca cola light de 3l retornable | 9982033 | 30.25 | 12000 |
| 8 | Coca Cola | Coca cola para taqueria de vidrio | 60066568 | 10.00 | 14800 |
| 9 | Atun premium | Atun de aceite marca dolores | 66050500 | 500.00 | 0 |
| 10 | Atun premium | Atun de agua marca dolores | 156304 | 500.00 | 10 |
+----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> selectr * from carro_compra;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'selectr * from carro_compra' at line 1
mysql> select * from carro_compra;
+----+-----+-----+
| id_compra | cantidad | id_articulo |
+----+-----+-----+
| 1 | 10 | 9 |
| 2 | 5 | 6 |
| 3 | 100 | 8 |
| 4 | 100 | 8 |
| 5 | 10 | 6 |
| 6 | 10 | 6 |
| 7 | 2 | 9 |
+----+-----+-----+
7 rows in set (0.00 sec)

mysql>

```

Figura 54: Base de datos con el contenido del carrito de compra.

4. Conclusiones

Al igual que en la tarea anterior observamos el funcionamiento de una aplicación REST, ahora esto lo podemos ver como un sistema mas competo y no solo eso y es que lo que se desarrollo es incluso un sistema que se podría vender a la típica tiendita de la esquina para que pueda llevar un control del inventario de forma electrónica y es que en la mayoría de las tiendas de la esquina aun siguen usando papel para llevar el inventariado de lo que venden y no solo lo podemos dejar aquí y es que si se le dedica mas tiempo a esta aplicación estamos hablando de poder desarrollar un e-commerce ya sea para que nosotros vendamos cosas o venderlo para que otras personas lo usen para sus fines personales, sin duda alguna una práctica interesante de desarrollar e integradora varias materias que se toman a lo largo de la carrera en ESCOM. Mencionar también que todo lo usado se encontrara en este mismo .zip por si se requiere hacer pruebas del trabajo entregado, también recordar que el script de la base de datos se encuentra disponible en el archivo .txt

Referencias

- Dev.mysql.com. 2021. MySQL :: MySQL Connector/J 5.1 Developer Guide :: 5.5 Java, JDBC and MySQL Types. [online] Available at: <https://dev.mysql.com/doc/connector-j/5.1/en/connector-j-reference-type-conversions.html> [Accessed 13 November 2021].
- Docs.oracle.com. 2021. PreparedStatement (Java Platform SE 7). [online] Available at: [https://docs.oracle.com/javase/7/docs/api/java/sql/PreparedStatement.html#setBigDecimal\(int,%20java.math.BigDecimal\)](https://docs.oracle.com/javase/7/docs/api/java/sql/PreparedStatement.html#setBigDecimal(int,%20java.math.BigDecimal)) [Accessed 13 November 2021].

Cis.upenn.edu. 2021. [online] Available at: <https://www.cis.upenn.edu/~bcpierce/courses/629/jdkdocs/guide/jdbc/getstart/mapping.doc.html> [Accessed 13 November 2021].