



**TECNOLÓGICO  
NACIONAL DE MÉXICO**



# **INSTITUTO NACIONAL DE MÉXICO**

## **INSTITUTO TÉCNOLOGICO DE TLAXIACO**

### **ACTIVIDAD**

#### **REPORTE DE PROYECTO INYECCIÓN DE CÓDIGO EN UNA TIENDA DE ABARROTES CON FINES**

#### **INTEGRANTES DE EQUIPO:**

- Saúl López Bautista
- Luz Arleth López Bautista

#### **DOCENTE:**

Ing. Edward Osorio Salinas

#### **GRADO:**

Séptimo semestre

#### **GRUPO:**

7US

## Contenido

Introducción .....	5
Propósito .....	5
Objetivos .....	5
Marco Teórico .....	5
Diseño del Sistema .....	6
1. Diseña una página web vulnerable a inyección SQL .....	7
2. Inyección de código SQL a campos específicos. ....	14
2.1 Inyección de código hacia la tabla Productos. ....	15
2.2 Inyección de código para eliminar sin autorización los registros en la tabla de Ventas Registradas. ....	19
2.3 Inyección de código para extraer los datos de los clientes. ....	21
3. Como prevenir las inyecciones SQL en nuestros códigos.....	23
3.1 Prevención en nuestro código Productos.php .....	23
3.2 Prevención en nuestro código Ventas.php .....	24
3.3 Prevención en nuestro código ver_clientes.php .....	25
Conclusión .....	26

## ilustraciones

Ilustración 1. Página principal de nuestra tienda de abarrotes .....	7
Ilustración 2. Listado de productos .....	7
Ilustración 3. Apartado de agregar un nuevo cliente .....	8
Ilustración 4 Apartado para el registro venta y tabla de ventas Registradas.....	9
Ilustración 5. Despliegue .....	9
Ilustración 6 Agregar Productos y unidades.....	10
Ilustración 7. Tabla de clientes.....	10
Ilustración 8 Registro.....	11
Ilustración 9. Datos almacenados .....	11
Ilustración 10 Tabla clientes .....	11
Ilustración 11. Registro de ventas .....	12
Ilustración 12. Registro exitoso de ventas.....	12
Ilustración 13. Registro de nuevo producto .....	13
Ilustración 14. Producto agregado .....	13
Ilustración 15. Registro de producto almacenado correctamente .....	13
Ilustración 16. Tabla listado de productos para inyección .....	14
Ilustración 17. Tabla Ventas Registradas para inyección .....	14
Ilustración 18. Tabla Listado de clientes para inyección .....	14
Ilustración 19. Código Vulnerable de productos.php.....	15
Ilustración 20. Código a modificar .....	15
Ilustración 21. Código modificado.....	15
Ilustración 22 Código completo de la inyección.....	16
Ilustración 23. Acceder a la URL .....	16
Ilustración 24. Campo para editar Producto .....	17
Ilustración 25. Dato ingresado por el atacante .....	17
Ilustración 26. Productos actualizados.....	19
Ilustración 27.Datos modificados y alterados .....	19
Ilustración 28. Datos de nuestra tabla .....	20
Ilustración 29. Código a inyectar .....	20
Ilustración 30. Código a modificar .....	20

Ilustración 31. Código modificado .....	20
Ilustración 32. Código modificado .....	20
Ilustración 33. Tabla eliminada por inyección .....	21
Ilustración 34. Tabla Clientes.....	21
Ilustración 35. Inyección de consulta .....	21
Ilustración 36. Opciones .....	22
Ilustración 37. Campo seleccionado a la consulta.....	22
Ilustración 38. Consulta adquirida.....	23
Ilustración 39. Código sanitizado.....	24
Ilustración 40. Modificación de consulta .....	24
Ilustración 41. Conversión de variables a enteros .....	24
Ilustración 42. Modificación de URL.....	24
Ilustración 43. Agregamos htmlspecialchars.....	25
Ilustración 44. Cambio de consulta .....	25
Ilustración 45. Agregamos intval .....	25
Ilustración 46. Inyección sin funcionamiento.....	25

## Introducción

La inyección de código es una técnica ampliamente utilizada en ciberataques para explotar vulnerabilidades en páginas web. Este proyecto busca demostrar cómo los desarrolladores pueden mitigar estos riesgos en sistemas utilizados por pequeñas empresas, como una tienda de abarrotes.

## Propósito

Concientizar sobre los riesgos de seguridad cibernética y promover buenas prácticas en el desarrollo de software seguro.

## Objetivos

- **General:** Simular y analizar vulnerabilidades por inyección de código en una tienda de abarrotes.
- **Específicos:**
  1. Diseñar una base de datos y una página web que simule un entorno real, mostrar que almacene.
  2. Implementar ataques de inyección SQL en campos específicos.
  3. Proponer soluciones para mitigar estas vulnerabilidades.

## Marco Teórico

La inyección de código, especialmente inyección SQL, ocurre cuando un atacante inserta código malicioso en un campo de entrada, comprometiendo la base de datos de una aplicación. Esto puede provocar:

- Robo de datos sensibles.
- Alteración o eliminación de datos.
- Ejecución de comandos administrativos.

En el caso de una tienda de abarrotes, una base de datos mal diseñada y sin validación adecuada en los formularios web podría exponer información confidencial de clientes, como su dirección y teléfono, o permitir manipular datos relacionados con productos y ventas.

## **Diseño del Sistema**

Se creó una base de datos con las siguientes tablas principales:

- agregar\_productos: Registra productos añadidos al inventario.
- clientes: Almacena información de clientes.
- detalle\_ventas: Detalla productos vendidos.
- productos: Lista de productos disponibles.
- ventas: Resumen de ventas realizadas.

## 1. Diseña una página web vulnerable a inyección SQL

Paso 1. Para crear nuestra interfaz principal de nuestra tienda de abarrotes hemos creado la siguiente interfaz que se muestra a continuación, nuestra interfaz está conformada por 5 opciones las cuales son ver productos, Agregar Cliente, Ver Ventas, Agregar Productos y Ver Clientes, cada una de las opciones nos lleva a diferentes apartados.



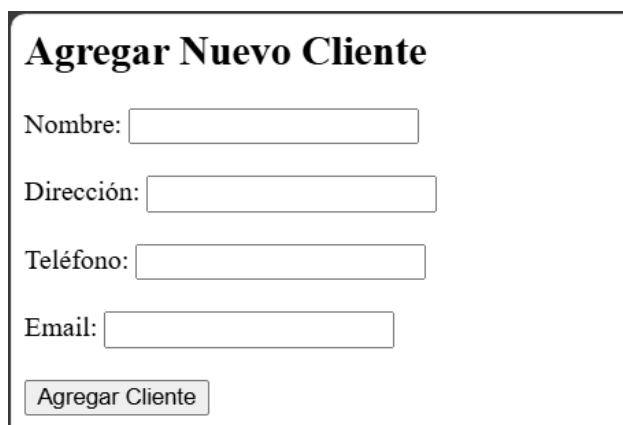
Ilustración 1. Página principal de nuestra tienda de abarrotes

Paso 2. Nos dirigimos a la primera opción la cual nos muestra una lista de productos que cuenta la tienda de abarrotes, al igual que las acciones que podemos realizar sobre cada una en dado caso que tengamos una inconveniencia a la hora de registro, donde podemos editar y eliminar

Listado de Productos					
ID	Nombre	Descripción	Precio	Stock	Acciones
2	Frijol Negro 1 kg	Frijol negro seleccionado para sopas y guarniciones.	28.40	40	<a href="#">Editar</a>   <a href="#">Eliminar</a>
3	Aceite Vegetal 1 L	Aceite vegetal para cocinar y freír.	35.00	30	<a href="#">Editar</a>   <a href="#">Eliminar</a>
4	Azúcar Refinada 1 kg	Azúcar blanca refinada para postres y bebidas.	20.00	45	<a href="#">Editar</a>   <a href="#">Eliminar</a>
5	Sal de Mesa 1 kg	Sal de mesa iodada en bolsa.	10.00	60	<a href="#">Editar</a>   <a href="#">Eliminar</a>
6	Harina de Trigo 1 kg	Harina de trigo para todo uso.	18.00	50	<a href="#">Editar</a>   <a href="#">Eliminar</a>
7	Leche Entera 1 L	Leche entera pasteurizada y homogeneizada.	22.00	35	<a href="#">Editar</a>   <a href="#">Eliminar</a>
8	Café Molido 500 g	Café molido de tueste medio.	75.00	25	<a href="#">Editar</a>   <a href="#">Eliminar</a>
9	Galletas de Mantequilla 200 g	Galletas tradicionales para acompañar el café.	25.00	30	<a href="#">Editar</a>   <a href="#">Eliminar</a>

Ilustración 2. Listado de productos

Paso 3. Después de haber visto los registros de los productos con los que cuenta la tienda de abarrotes pasamos a la siguiente opción el cual es registrar un cliente.



**Agregar Nuevo Cliente**

Nombre:

Dirección:

Teléfono:

Email:

**Ilustración 3. Apartado de agregar un nuevo cliente**



Paso 4. Esta es la interfaz de la opción de ver ventas donde Nos permitirá agregar las ventas que sean realizadas por nuestros clientes en este caso en la parte derecha donde este el apartado de clientes nos aparece una opción para seleccionar en este caso no nos muestra ningún nombre ya que la base de datos aún no tiene almacenado los nombres de los clientes, en la siguiente opción que es Producto podemos observar que nos aparece el nombre de Frijol Negro 1 kg el cual es tomado de la lista de nuestros producto, si desplegamos esa barra nos aparecerán más producto con los cuales cuenta la tienda , al igual que la cantidad de producto que adquirió el cliente, en el apartado que dice Ventas Registradas nos mostrara una tabla que contendrá las ventas registradas, en este caso no hemos almacenado nada aun.



**Registrar Venta**

Cliente:

Producto:

Cantidad:

**Ventas Registradas**

No hay ventas registradas.

© 2024 Tienda de Abarrotes

Ilustración 4 Apartado para el registro venta y tabla de ventas Registradas



**Registrar Venta**

Cliente:

Producto:

Cantidad:

**Ventas**

No hay ve

© 2024 Ti

- Frijol Negro 1 kg
- Aceite Vegetal 1 L
- Azúcar Refinada 1 kg
- Sal de Mesa 1 kg
- Harina de Trigo 1 kg
- Leche Entera 1 L
- Café Molido 500 g
- Galletas de Mantequilla 200 g

Ilustración 5. Despliegue

Paso 5. En la siguiente ventana que es Agregar productos, es la primordial que nos sirve para agregar los productos en nuestra tabla que se mostró el inicio donde veíamos un enlistado de productos al igual que hemos creado dentro del mismo sitio Agregar Unidades a un Producto Existente, el cual nos sirve para seleccionar algún producto de nuestra lista que tenemos y agregarle más cantidad, por ejemplo si en la tabla de productos tenemos Frijol Negro con una cantidad de 5 al seleccionarlo y en el apartado que dice cantidad a agregar le pones un 9 entonces en nuestra tabla ya no estaría 5 sino 14 como cantidad.

The image shows two web forms. The first form, titled 'Agregar Nuevo Producto', has input fields for 'Nombre del Producto:', 'Descripción:', 'Precio:', and 'Stock Inicial:', followed by an 'Agregar Producto' button. The second form, titled 'Agregar Unidades a un Producto Existente', has a dropdown menu for 'Selecciona un Producto:' with 'Frijol Negro 1 kg' selected, an input field for 'Cantidad a Agregar:', and an 'Agregar Unidades' button.

**Ilustración 6 Agregar Productos y unidades**

Paso 6. Una vez terminado de a ver visto la opción anterior nos dirigimos a la ultima la cual sería Ver Clientes el cual mostrará la tabla que contendrá a nuestros clientes, en este caso no hemos registrado ningún cliente aun.

The image shows a web page titled 'Listado de Clientes'. Below the title, it says 'No se encontraron clientes registrados.' Below this text is a table with the following headers: ID, Nombre, Dirección, Teléfono, Email, and Acciones.

ID	Nombre	Dirección	Teléfono	Email	Acciones
No se encontraron clientes registrados.					

**Ilustración 7. Tabla de clientes**

Paso 7. Una vez terminado de a ver mostrado cada uno de los apartados de nuestras opciones proseguimos a el almacenamiento de datos para después proseguir con las inyecciones SQL, agregamos los siguientes datos correspondientes

## Agregar Nuevo Cliente

Nombre: Luz Arleth López Bautista

Dirección: oaxaca

Teléfono: 9531613220

Email: lopezbautistasaul12@gmail

Agregar Cliente

### Ilustración 8 Registro

Nuevo cliente agregado con éxito!

## Agregar Nuevo Cliente

Nombre:

Dirección:

Teléfono:

Email:

Agregar Cliente

### Ilustración 9. Datos almacenados

Paso 8. Nos dirigimos a nuestro apartado ver clientes, para verificar que los datos si, se estén mostrando, podemos observar que los datos ingresados si se muestran

ID	Nombre	Dirección	Teléfono	Email	Acciones
13	Luz Arleth López Bautista	oaxaca	9531613220	lopezbautistasaul12@gmail.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>

### Ilustración 10 Tabla clientes

Paso 9. Proseguimos a realizar el siguiente registro en el apartado de ventas, ingresamos los registros por default nos da el único nombre que hemos almacenado en este caso es Luz Arleth López Bautista, que adquirido como producto frijol negro con una cantidad de 5, damos clic en el botón que dice Registrar venta, se puede apreciar que el registro fue exitosamente almacenado al igual se muestra su respectiva tabla.

## Registrar Venta

Cliente:

Luz Arleth López Bautista ▾

Producto:

Frijol Negro 1 kg ▾

Cantidad:

5

Registrar Venta

## Ventas Registradas

No hay ventas registradas.

© 2024 Tienda de Abarrotes

Ilustración 11. Registro de ventas

Venta registrada exitosamente!

## Registrar Venta

Cliente:

Luz Arleth López Bautista ▾

Producto:

Frijol Negro 1 kg ▾

Cantidad:

Registrar Venta

## Ventas Registradas

ID Venta	Cliente	Producto	Cantidad	Precio	Total	Acciones
14	Luz Arleth López Bautista	Frijol Negro 1 kg	5	28.40	142.00	<a href="#">Editar</a>   <a href="#">Eliminar</a>

© 2024 Tienda de Abarrotes

Ilustración 12. Registro exitoso de ventas

Paso 10. De igual forma con agregar un producto, en este caso registramos un producto que es Sardina, proseguimos a darle clic en Agregar Producto

### Agregar Nuevo Producto

Nombre del Producto:

Descripción:

Precio:

Stock Inicial:

Ilustración 13. Registro de nuevo producto

Nuevo producto agregado con éxito!

### Agregar Nuevo Producto

Nombre del Producto:

Descripción:

Precio:

Stock Inicial:

Ilustración 14. Producto agregado

### Listado de Productos

ID	Nombre	Descripción	Precio	Stock	Acciones
2	Frijol Negro 1 kg	Frijol negro seleccionado para sopas y guarniciones.	28.40	106	<a href="#">Editar</a>   <a href="#">Eliminar</a>
3	Aceite Vegetal 1 L	Aceite vegetal para cocinar y freír.	35.00	30	<a href="#">Editar</a>   <a href="#">Eliminar</a>
4	Azúcar Refinada 1 kg	Azúcar blanca refinada para postres y bebidas.	20.00	45	<a href="#">Editar</a>   <a href="#">Eliminar</a>
5	Sal de Mesa 1 kg	Sal de mesa iodada en bolsa.	10.00	60	<a href="#">Editar</a>   <a href="#">Eliminar</a>
6	Harina de Trigo 1 kg	Harina de trigo para todo uso.	18.00	50	<a href="#">Editar</a>   <a href="#">Eliminar</a>
7	Leche Entera 1 L	Leche entera pasteurizada y homogeneizada.	22.00	35	<a href="#">Editar</a>   <a href="#">Eliminar</a>
8	Café Molido 500 g	Café molido de tueste medio.	75.00	25	<a href="#">Editar</a>   <a href="#">Eliminar</a>
9	Galletas de Mantequilla 200 g	Galletas tradicionales para acompañar el café.	25.00	30	<a href="#">Editar</a>   <a href="#">Eliminar</a>
11	Sardina	Un rico y nutritivo sazón	30.00	11	<a href="#">Editar</a>   <a href="#">Eliminar</a>

Ilustración 15. Registro de producto almacenado correctamente

## 2. Inyección de código SQL a campos específicos.

Paso 1. Hemos agregado más datos a los campos que estaremos vulnerando para realizar nuestras inyecciones sql a continuación se muestran las tablas que serán atacadas.

**Listado de Productos**

ID	Nombre	Descripción	Precio	Stock	Acciones
2	Frijol Negro 1 kg	Frijol negro seleccionado para sopas y guarniciones.	28.40	106	<a href="#">Editar</a>   <a href="#">Eliminar</a>
3	Aceite Vegetal 1 L	Aceite vegetal para cocinar y freír.	35.00	30	<a href="#">Editar</a>   <a href="#">Eliminar</a>
4	Azúcar Refinada 1 kg	Azúcar blanca refinada para postres y bebidas.	20.00	45	<a href="#">Editar</a>   <a href="#">Eliminar</a>
5	Sal de Mesa 1 kg	Sal de mesa iodada en bolsa.	10.00	60	<a href="#">Editar</a>   <a href="#">Eliminar</a>
6	Harina de Trigo 1 kg	Harina de trigo para todo uso.	18.00	50	<a href="#">Editar</a>   <a href="#">Eliminar</a>
7	Leche Entera 1 L	Leche entera pasteurizada y homogeneizada.	22.00	35	<a href="#">Editar</a>   <a href="#">Eliminar</a>
8	Café Molido 500 g	Café molido de tueste medio.	75.00	25	<a href="#">Editar</a>   <a href="#">Eliminar</a>
9	Galletas de Mantequilla 200 g	Galletas tradicionales para acompañar el café.	25.00	30	<a href="#">Editar</a>   <a href="#">Eliminar</a>
11	Sardina	Un rico y nutritivo sazón	30.00	11	<a href="#">Editar</a>   <a href="#">Eliminar</a>

**Ilustración 16. Tabla listado de productos para inyección**

**Ventas Registradas**

ID Venta	Cliente	Producto	Cantidad	Precio	Total	Acciones
14	Luz Arleth	Frijol Negro 1 kg	5	28.40	142.00	<a href="#">Editar</a>   <a href="#">Eliminar</a>
15	Maria Gómez	Harina de Trigo 1 kg	3	18.00	54.00	<a href="#">Editar</a>   <a href="#">Eliminar</a>
16	Juan Pérez	Sal de Mesa 1 kg	2	10.00	20.00	<a href="#">Editar</a>   <a href="#">Eliminar</a>
17	Laura Rodríguez	Leche Entera 1 L	2	22.00	44.00	<a href="#">Editar</a>   <a href="#">Eliminar</a>
18	Andrés Sánchez	Azúcar Refinada 1 kg	3	20.00	60.00	<a href="#">Editar</a>   <a href="#">Eliminar</a>
19	Patricia Díaz	Harina de Trigo 1 kg	2	18.00	36.00	<a href="#">Editar</a>   <a href="#">Eliminar</a>
20	Carlos López	Galletas de Mantequilla 200 g	3	25.00	75.00	<a href="#">Editar</a>   <a href="#">Eliminar</a>

**Ilustración 17. Tabla Ventas Registradas para inyección**

**Listado de Clientes**

ID	Nombre	Dirección	Teléfono	Email	Acciones
13	Luz Arleth	oaxaca	9531613220	lopezbautistasaul12@gmail.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>
14	Juan Pérez	Calle 123, Ciudad X	555-123-4567	juan.perez@gmail.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>
15	Maria Gómez	Av. Central 456	555-234-5678	maria.gomez@gmail.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>
16	Carlos López	Calle Ficticia 789	555-345-6789	carlos.lopez@gmail.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>
17	Laura Rodríguez	Plaza Mayor 321	555-456-7890	laura.rodriguez@outklok.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>
18	Andrés Sánchez	Calle 101, Colonia B	555-567-8901	andres.sanchez@gmail.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>
19	Patricia Díaz	Av. Libertad 654	555-678-9012	patricia.diaz@gmail.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>

**Ilustración 18. Tabla Listado de clientes para inyección**

## 2.1 Inyección de código hacia la tabla Productos.

Paso 2. Nos dirigimos a nuestro archivo productos.php para realizar la inyección sql vemos que nuestro parámetro id\_producto lo podemos inyectar con 1 OR 1=1 – accediendo desde la URL con esos valores.

```
<?php
include 'conexion.php';

$sql = "SELECT * FROM productos";
$result = $conn->query($sql);

echo "<h2>Listado de Productos</h2>";
echo "<table border='1'>
<tr>
<th>ID</th>
<th>Nombre</th>
<th>Descripción</th>
<th>Precio</th>
<th>Stock</th>
<th>Acciones</th> <!-- Columna para los botones de editar y eliminar -->
</tr>";

if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo "<tr>
        <td>" . $row["id_producto"] . "</td>
        <td>" . $row["nombre_producto"] . "</td>
        <td>" . $row["descripcion"] . "</td>
        <td>" . $row["precio"] . "</td>
        <td>" . $row["stock"] . "</td>
        <td>
        <a href='editar_producto.php?id_producto=" . $row["id_producto"] . "'>Editar</a> |
        <a href='eliminar_producto.php?id_producto=" . $row["id_producto"] . "' onclick='return confirm(\"¿Estás seguro de que deseas e
        </td>";
    }
    echo "</table>";
} else {
    echo "0 resultados";
}

$conn->close();
```

Ilustración 19. Código Vulnerable de productos.php

Paso 3. Modificamos esta línea de código.

```
<a href='editar_producto.php?id_producto=" . $row["id_producto"] . "'>Editar</a> |
```

Ilustración 20. Código a modificar

Por esta.

```
<a href='http://localhost/tienda_abarrotes/editar_producto.php?id_producto=1%20OR%201=1%20--'>Editar</a>
```

Ilustración 21. Código modificado

#### Paso 4. Explicación del código modificado

`<a href="http://localhost/tienda_abarrotes/editar_producto.php?id_producto=1%20OR%201=1%20--">Editar</a>`, lo que quiere decir que se accedera al localhost, ingresando a la carpeta del proyecto en nuestro directorio raíz del servidor accediendo al archivo `editar_producto.php`, donde tenemos el parámetros `id_producto` de nuestro archivo `productos` se ingresó `id_producto=1%20OR%201=1%20`—lo que hace es inyectar con los espacios reemplazados por `%20` (la codificación URL). El código `OR 1=1` —esta parte es la inyección por lo que se vería así el código completo.

```
include 'conexion.php';

$sql = "SELECT * FROM productos";
$result = $conn->query($sql);

echo "<h2>Listado de Productos</h2>";
echo "<table border='1'>";
<tr>
    <th>ID</th>
    <th>Nombre</th>
    <th>Descripción</th>
    <th>Precio</th>
    <th>Stock</th>
    <th>Acciones</th> <!-- Columna para los botones de editar y eliminar -->
</tr>";

if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo "<tr>";
        <td>" . $row["id_producto"] . "</td>
        <td>" . $row["nombre_producto"] . "</td>
        <td>" . $row["descripcion"] . "</td>
        <td>" . $row["precio"] . "</td>
        <td>" . $row["stock"] . "</td>
        <td>
            <a href='http://localhost/tienda_abarrotes/editar_producto.php?id_producto=1%20OR%201=1%20--'>Editar</a>
            <a href='eliminar_producto.php?id_producto=" . $row["id_producto"] . "' onclick='return confirm(\"¿Estás seguro de que deseas e
        </td>
    </tr>";
    }
    echo "</table>";
} else {
    echo "0 resultados";
}
```

Ilustración 22 Código completo de la inyección

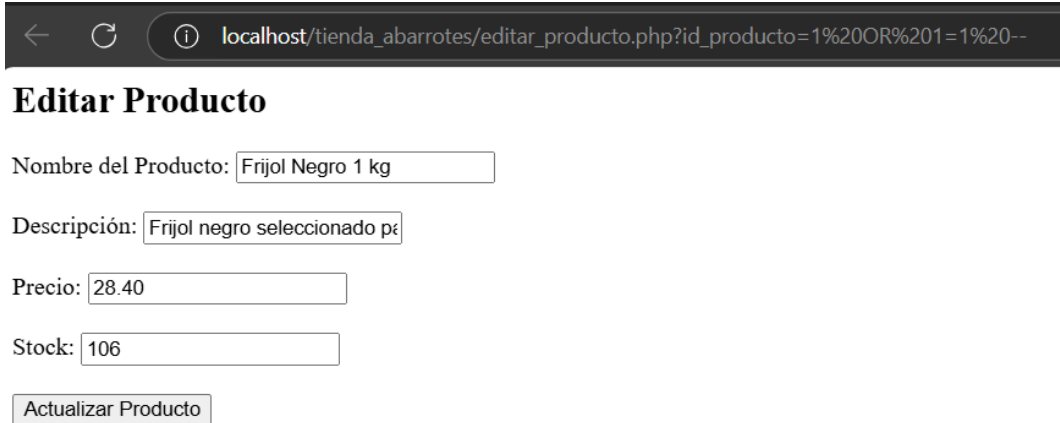
Paso 5. Ponemos nuestro cursor en la URL, podemos observar que el atacante podrá acceder al vínculo de nuestro archivo, damos clic.

```
<td>" . $row["precio"] . "</td>
<td>" . $row["
<td>
    <a href='http://http://localhost/tienda_abarrotes/editar_producto.php?id_producto=1%20OR%201=1%20-- %201=1%
    <a href='eliminar_producto.php?id_producto=" . $row["id_producto"] . "' onclick='return
</td>
```

Ilustración 23. Acceder a la URL



Paso 6. Podemos notar que en nuestra dirección URL está totalmente corrompido por la inyección SQL, al igual que el atacante pudo acceder a realizar el respectivo cambio.

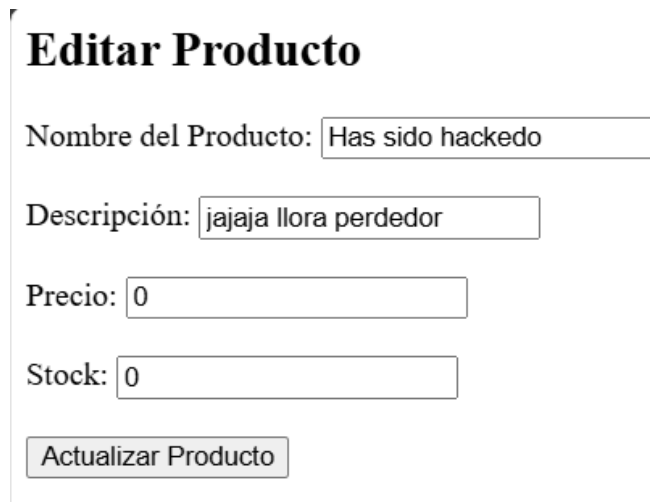


The screenshot shows a web browser window with a dark address bar. The URL is `localhost/tienda_abarrotes/editar_producto.php?id_producto=1%20OR%201=1%20--`. Below the address bar, the page title is "Editar Producto". The form contains the following fields:

- Nombre del Producto:
- Descripción:
- Precio:
- Stock:
- 

**Ilustración 24. Campo para editar Producto**

Paso 7. Realizamos los respectivos cambios el cual quedaría así como se muestra en la imagen, damos clic en Actualizar Producto.



The screenshot shows the "Editar Producto" form after the attack. The fields now contain the following values:

- Nombre del Producto:
- Descripción:
- Precio:
- Stock:
- 

**Ilustración 25. Dato ingresado por el atacante**

Paso 8. Una vez dándole clic al botón se puede apreciar que los datos fueron actualizados correctamente.

Producto actualizado con éxito!

**Editar Producto**

Nombre del Producto:

Descripción:

Precio:

Stock:

**Ilustración 26. Productos actualizados**

Paso 9. Nos pasamos a nuestro apartado de la opción de Productos para ver nuestra tabla de como ha quedado, podemos ver que toda la tabla fue registrada por el atacante, donde se ha perdido todo el registro que contenía nuestra tabla.

## Listado de Productos

ID	Nombre	Descripción	Precio	Stock	Acciones
2	Has sido hackeado	jajaja lora perdedor	0.00	0	<a href="#">Editar</a> <a href="#">Eliminar</a>
3	Has sido hackeado	jajaja lora perdedor	0.00	0	<a href="#">Editar</a> <a href="#">Eliminar</a>
4	Has sido hackeado	jajaja lora perdedor	0.00	0	<a href="#">Editar</a> <a href="#">Eliminar</a>
5	Has sido hackeado	jajaja lora perdedor	0.00	0	<a href="#">Editar</a> <a href="#">Eliminar</a>
6	Has sido hackeado	jajaja lora perdedor	0.00	0	<a href="#">Editar</a> <a href="#">Eliminar</a>
7	Has sido hackeado	jajaja lora perdedor	0.00	0	<a href="#">Editar</a> <a href="#">Eliminar</a>
8	Has sido hackeado	jajaja lora perdedor	0.00	0	<a href="#">Editar</a> <a href="#">Eliminar</a>
9	Has sido hackeado	jajaja lora perdedor	0.00	0	<a href="#">Editar</a> <a href="#">Eliminar</a>
11	Has sido hackeado	jajaja lora perdedor	0.00	0	<a href="#">Editar</a> <a href="#">Eliminar</a>

Ilustración 27. Datos modificados y alterados

## 2.2 Inyección de código para eliminar sin autorización los registros en la tabla de Ventas Registradas.

Paso 1 nos dirigimos a nuestro apartado ver ventas para observar lo que tenemos almacenado el cual es fundamental ya que nos permite dar a conocer los distintos clientes que adquieren nuestro productos.

### Ventas Registradas

ID Venta	Cliente	Producto	Cantidad	Precio	Total	Acciones
21	Luz Arleth	Frijol Negro 1 kg	4	20.00	80.00	<a href="#">Editar</a> <a href="#">Eliminar</a>
22	Juan Pérez	Aceite Vegetal 1 L	3	35.00	105.00	<a href="#">Editar</a> <a href="#">Eliminar</a>
23	Maria Gómez	Azúcar Refinada 1 kg	5	20.00	100.00	<a href="#">Editar</a> <a href="#">Eliminar</a>
24	Carlos López	Sal de Mesa 1 kg	5	10.00	50.00	<a href="#">Editar</a> <a href="#">Eliminar</a>
25	Laura Rodríguez	Leche Entera 1 L	3	22.00	66.00	<a href="#">Editar</a> <a href="#">Eliminar</a>
26	Andrés Sánchez	Café Molido 500 g	3	75.00	225.00	<a href="#">Editar</a> <a href="#">Eliminar</a>
27	Patricia Díaz	Galletas de Mantequilla 200 g	6	25.00	150.00	<a href="#">Editar</a> <a href="#">Eliminar</a>

Ilustración 28. Datos de nuestra tabla

Paso 2. Nos dirigimos al archivo ventas.php y modificamos de igual forma el id\_venta pero en este caso se lo aplicaremos a la sección de eliminar.

```

        <th>Cantidad</th>
        <th>Precio</th>
        <th>Total</th>
        <th>Acciones</th>
    </tr>";
    while ($row = $resultVentas->fetch_assoc()) {
        echo "<tr>";
        <td> . $row['id_venta'] . "</td>
        <td> . $row['cliente'] . "</td>
        <td> . $row['producto'] . "</td>
        <td> . $row['cantidad'] . "</td>
        <td> . $row['precio'] . "</td>
        <td> . $row['total'] . "</td>
        <td>
            <a href='editar_venta.php?id_venta=' . $row['id_venta'] . "'>Editar</a> |
            <a href='ventas.php?eliminar=' . $row['id_venta'] . "' onclick='return confirm(\"¿Seguro que deseas eliminar esta v
        </td>
    </tr>";
    }
    echo "</table>";
} else {
    echo "No hay ventas registradas.";
}

```

Ilustración 29. Código a inyectar

Paso 3. Este es el apartado que modificaremos

```
<a href='ventas.php?eliminar=' . $row['id_venta'] . "' onclick='return confirm(\"¿Seguro que deseas eliminar esta v
```

Ilustración 30. Código a modificar

Y sustituiremos por:

```
<a href='http://localhost/ventas.php?eliminar=1 OR 1=1>Eliminar</a>
```

Ilustración 31. Código modificado

Paso 4. Ponemos el cursor en el enlace que hemos agregado, al darle clic nos mandara automáticamente a la interfaz donde tenemos los registros pero en este caso no aparecerá nada ya que todo será eliminado por la inyección de código sql.

```
<a href='Seguir vínculo (ctrl + clic)'ta=" . $row['id_venta'] . "'>Editar</a> |
<a href='http://localhost/ventas.php?eliminar=1 OR 1=1>Eliminar</a>
```

Ilustración 32. Código modificado

Paso 5. Podemos notar que la tabla fue eliminada automáticamente.

← ↻ ⓘ localhost/tienda\_abarrotes/ventas.php?eliminar=1%20OR%201=1

Venta eliminada exitosamente!

### Registrar Venta

Cliente: Luz Arleth

Producto: Frijol Negro 1 kg

Cantidad:

Registrar Venta

### Ventas Registradas

No hay ventas registradas.

© 2024 Tienda de Abarrotes

Ilustración 33. Tabla eliminada por inyección

### 2.3 Inyección de código para extraer los datos de los clientes.

Paso 1. Nos dirigimos a nuestro apartado de ver clientes para ver los datos que almacena nuestra tabla.

Listado de Clientes

ID	Nombre	Dirección	Teléfono	Email	Acciones
13	Luz Arleth	oaxaca	9531613220	lopezbautistasaul12@gmail.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>
14	Juan Pérez	Calle 123, Ciudad X	555-123-4567	juan.perez@gmail.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>
15	Maria Gómez	Av. Central 456	555-234-5678	maria.gomez@gmail.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>
16	Carlos López	Calle Ficticia 789	555-345-6789	carlos.lopez@gmail.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>
17	Laura Rodríguez	Plaza Mayor 321	555-456-7890	laura.rodriguez@outklok.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>
18	Andrés Sánchez	Calle 101, Colonia B	555-567-8901	andres.sanchez@gmail.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>
19	Patricia Díaz	Av. Libertad 654	555-678-9012	patricia.diaz@gmail.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>
20	pedro	Jalisco	9531613228	lopezbautistasaul12@gmail.com	<a href="#">Editar</a>   <a href="#">Eliminar</a>

Ilustración 34. Tabla Clientes

Paso 2. Una vez observando nuestra tabla nos dirigimos a nuestro código para realizar sus respectivos cambios.

```
// Simulación de inyecciones SQL
echo "<h2>Simulación de Inyección SQL</h2>";
echo "<a href='?id_cliente=1'>Consulta válida</a><br>";
echo "<a href='?id_cliente=1 OR 1=1'>Consulta inyectada</a><br>";

// Cerrar la conexión
$conn->close();*/
```

Ilustración 35. Inyección de consulta

Paso 3. Nos dirigimos a nuestra página para observar los cambios que se realizaron, podemos notar que nos aparecen dos opciones una con el nombre de consultad válida y la otra con consulta inyectada, damos clic en la primera opción

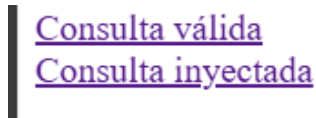


Ilustración 36. Opciones

Paso 4. Una vez dándole clic a la primera opción esto nos aparecerá, donde podemos notar que nuestra consulta al querer ver los datos fueron modificados, donde solo se puede notar el campo que fue llamado, en este caso nos muestra solo el id 17 el cual muestra los datos de Laura, sin mostrar las demás tablas.

```
Consulta ejecutada: SELECT * FROM clientes WHERE id_cliente = 17  
ID: 17  
Nombre: Laura Rodríguez  
Dirección: Plaza Mayor 321  
Teléfono: 555-456-7890  
Email: laura.rodriguez@outklok.com
```

Ilustración 37. Campo seleccionado a la consulta.

Paso 5. Nos dirigimos a la segunda opción que nos aparecía en pantalla y podemos observar que la consulta que no nos mostraba nada fue inyectada por los valores 1 OR 1 = 1 donde podemos notar que nuestra consulta muestra completamente los datos que hay en nuestra tabla

```

Consulta ejecutada: SELECT * FROM clientes WHERE id_cliente = 1 OR 1=1

ID: 13
Nombre: Luz Arleth
Dirección: oaxaca
Teléfono: 9531613220
Email: lopezbautistasaul12@gmail.com

ID: 14
Nombre: Juan Pérez
Dirección: Calle 123, Ciudad X
Teléfono: 555-123-4567
Email: juan.perez@gmail.com

ID: 15
Nombre: Maria Gómez
Dirección: Av. Central 456
Teléfono: 555-234-5678
Email: maria.gomez@gmail.com

ID: 16
Nombre: Carlos López
Dirección: Calle Ficticia 789
Teléfono: 555-345-6789
Email: carlos.lopez@gmail.com

ID: 17
Nombre: Laura Rodríguez
Dirección: Plaza Mayor 321
Teléfono: 555-456-7890
Email: laura.rodriguez@outklok.com

ID: 18
Nombre: Andrés Sánchez
Dirección: Calle 101, Colonia B
Teléfono: 555-567-8901
Email: andres.sanchez@gmail.com

ID: 19
Nombre: Patricia Díaz

```

**Ilustración 38. Consulta adquirida**

### 3. Como prevenir las inyecciones SQL en nuestros códigos.

#### 3.1 Prevención en nuestro código Productos.php

Paso 1. Reemplazamos el valor estático `id_producto=1%20OR%201=1%20--` con el valor dinámico `$row["id_producto"]` de manera segura utilizando `urlencode()`.

- Escapar caracteres especiales en los datos con `htmlspecialchars()` para evitar ataques XSS.
- Codificar los parámetros de la URL con `urlencode()` para evitar manipulaciones en los enlaces.
- Evitar inyección SQL en los enlaces de edición y eliminación al usar valores obtenidos de la base de datos de forma segura.

```

if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        // Sanitizar datos de salida para prevenir XSS
        $id_producto = htmlspecialchars($row["id_producto"]);
        $nombre_producto = htmlspecialchars($row["nombre_producto"]);
        $descripcion = htmlspecialchars($row["descripcion"]);
        $precio = htmlspecialchars($row["precio"]);
        $stock = htmlspecialchars($row["stock"]);

        echo "<tr>
            <td>$id_producto</td>
            <td>$nombre_producto</td>
            <td>$descripcion</td>
            <td>$precio</td>
            <td>$stock</td>
            <td>
                <!-- Usar enlaces seguros con id_producto dinámico -->
                <a href='editar_producto.php?id_producto=' . urlencode($id_producto) . "'>Editar</a>
                <a href='eliminar_producto.php?id_producto=' . urlencode($id_producto) . "' onclick='return confirm(\"¿Estás seguro de que deseas eliminar este producto?\");'>Eliminar</a>
            </td>
        </tr>";
    }
}

```

**Ilustración 39. Código sanitizado**

## 3.2 Prevención en nuestro código Ventas.php

Paso 1. Reemplazamos todas las consultas SQL dinámicas con consultas preparadas para evitar inyecciones SQL.

```

$stmt = $conn->prepare("SELECT precio FROM productos WHERE id_producto = ?");
$stmt->bind_param("i", $id_producto);

```

**Ilustración 40. Modificación de consulta**

Paso 2. Convertimos las variables de entrada como id\_cliente, id\_producto, y cantidad a enteros con intval() para asegurar que solo se procesen datos válidos.

```

// Usar prepared statements para evitar SQLi
$id_cliente = intval($_POST['id_cliente']);

```

**Ilustración 41. Conversión de variables a enteros**

Paso 3. Modifiqué el enlace de eliminación para evitar inyecciones SQL al pasar parámetros de URL.

```

<a href='ventas.php?eliminar=' . intval($row['id_venta']) . "'>Eliminar</a>

```

**Ilustración 42. Modificación de URL**

Paso 4. Agregamos htmlspecialchars() para escapar los datos antes de mostrarlos en el frontend, previniendo inyecciones XSS.



```
htmlspecialchars($row['nombre']) . "</option>";
```

Ilustración 43. Agregamos htmlspecialchars

### 3.3 Prevención en nuestro código ver\_clientes.php

Paso 1. Hemos cambiado el tipo de consulta a realizar aunque sigue mostrando información, es mejor evitar exponer los demás datos.

```
echo "<p><strong>Consulta ejecutada:</strong> SELECT * FROM clientes WHERE id_cliente = $id_cliente</p>";
```

Ilustración 44. Cambio de consulta

Paso 2. Aunque no cambia funcionalidad, el código ahora no permite ejecutar inyecciones por el uso de intval() y consultas preparadas.

```
$id_cliente = intval($_GET['id_cliente']);
```

Ilustración 45. Agregamos intval

```
echo "<a href='?id_cliente=17'>Consulta válida</a><br>";  
echo "<a href='?id_cliente=1 OR 1=1'>Consulta protegida</a><br>";
```

Ilustración 46. Inyección sin funcionamiento

## Conclusión

En conclusión, la inyección SQL es una de las vulnerabilidades más críticas y comunes en las aplicaciones web, que permite a los atacantes ejecutar consultas maliciosas y obtener acceso no autorizado a bases de datos, lo que puede tener consecuencias devastadoras para la seguridad de una organización. Los tipos de inyección SQL, como la basada en errores, la ciega, la de unión o la de tiempo, entre otras, muestran la diversidad de técnicas que los atacantes pueden utilizar para explotar sistemas mal diseñados o mal configurados.

Es fundamental implementar prácticas de seguridad sólidas, como el uso de consultas preparadas y parámetros bind, la validación y sanitización adecuada de las entradas, y la configuración de privilegios mínimos en las bases de datos. Además, es necesario ocultar errores del servidor para evitar que los atacantes obtengan información valiosa sobre la estructura interna de la base de datos. La protección contra la inyección SQL no solo mejora la seguridad, sino que también fortalece la confianza de los usuarios en los sistemas que emplean estas medidas preventivas.

En última instancia, la seguridad debe ser considerada desde las primeras etapas de desarrollo de una aplicación, no como una acción correctiva, sino como una parte integral de la arquitectura y diseño del sistema.