Informe Guerra de Testers

Chavez Salas Nadia Yunorvi, Eguía Lozada André Raúl, Pinto Ticona Nicole Alexandra, Valverde Riveros María Fernanda, Zuñiga Huanca Alexandra Yamiletti

Universidad Católica de Santa María, Arequipa, Perú

***Resumen*—En este informe se presentarán los distintos bugs o defectos encontrados en un sistema de facturación. Para ello se llevó a cabo el Test de Caja Negra, donde se realizaron pruebas siguiendo los requerimientos funcionales y no funcionales establecidos. Además, se incluirán observaciones más allá de los requisitos existentes.**

Se hará una matriz de trazabilidad para vincular los requisitos y evaluar si se cumplen las expectativas establecidas en la primera versión. Luego tendremos una sección donde mostraremos los requisitos separados por módulos, es decir detallaremos lo que debería de hacer cada uno. Seguidamente encontraremos varias tablas que nos indicará específicamente cuáles fueron los bugs encontrados al usar este sistema, esto coincide con los requerimientos funcionales que mencionamos en la sección anterior.

Finalmente, se realizará un análisis de los requisitos no funcionales y cómo se manifiestan al utilizar el sistema en tiempo real.

I. Introducción

Testing es una disciplina fundamental en el desarrollo de software, que tiene como objetivo evaluar la calidad y el funcionamiento del sistema o aplicación [1]. Consiste en ejecutar el software con el propósito de descubrir errores, defectos o fallas, así como verificar que cumple con los requisitos y expectativas establecidos. En este documento se ejecutarán técnicas de Testing con el fin de encontrar fallos en el software desarrollado por el equipo de trabajo asignado. Se hizo una matriz de trazabilidad para especificar los requerimientos, los estados y características actuales en el software, evaluando su desempeño en el mismo. Se escogió el Test de Caja Negra, por la carencia de documentación del código, haciendo un listado de funciones e interfaces para organizar el Test. Se añadió una descripción detallada de los casos, con el fin de poder ubicar con más exactitud los casos de Test, proporcionando sus resultados, adicionalmente como se implementaron estos casos de Test. Por último, se ha realizado un análisis de los requisitos no funcionales, identificando en detalle qué aspectos de los requisitos no se cumplen en el software proporcionado.

II. Observaciones

- La seguridad de la aplicación es inexistente ya que no nos pide un login previo para acceder o ingresar con algún código siendo este un requisito funcional.
- La base de datos no cuenta con ningún tipo de Procedimientos Almacenados, ni Triggers.
- La base de datos no se encuentra en un servidor en línea por lo que no se cumplió el requisito mínimo del módulo y no se pueden realizar más casos de prueba
- Al momento de ocurrir un fallo, no existe manejo de errores adecuado. Esto se puede apreciar, por ejemplo, al ingresar un carácter no numérico en el número de teléfono del Comprador
- Toda la logística de la aplicación se realizó con HTML siendo así que no se pudieron aplicar test de acuerdo a lo enseñado en clase.

III. MATRIZ DE TRAZABILIDAD

IDENTIFICACION		ESTADO		CARACTERISTICAS	
ID	**DESCRIPCION**	**Satisfecho**	**VERSION**	**RESPONSABLE**	**PRIORIDAD**
RF01	Registro de clientes: El sistema debe permitir registrar los datos personales y de contacto de los clientes.	SI	1.0	Equipo 2	No Especificada
RF02	Gestión de productos:El sistema debe permitir registrar los productos que se venden en la farmacia, incluyendo su nombre, descripción, precio y cantidad.	SI	1.0	Equipo 2	No Especificada
RF03	Generación de factura: El sistema debe permitir generar facturas para los clientes que realicen compras. Las facturas deben incluir los detalles de los productos comprados, el precio y la cantidad.	NO	1.0	Equipo 2	No Especificada
RF04	Cálculo de impuestos: El sistema debe calcular automáticamente los impuestos aplicables a cada compra y mostrarlos en la factura.	NO	1.0	Equipo 2	No Especificada
RF05	Gestión de inventario:El sistema debe permitir llevar un registro del inventario de la farmacia, incluyendo la cantidad de cada producto disponible y las fechas de vencimiento.	NO	1.0	Equipo 2	No Especificada
RF06	Gestión de pagos:El sistema debe permitir registrar los pagos de los clientes y llevar un registro de las transacciones.	SI	1.0	Equipo 2	No Especificada
REQUISITOS NO FUNCIONALES					
RNF01	Usabilidad: El sistema debe ser fácil de usar y comprender para los empleados de la farmacia	SI	1.0	Equipo 2	No Especificada
RNF02	Eficiencia: El sistema debe ser capaz de procesar las transacciones de forma rápida y eficiente, sin retrasos notables.	SI	1.0	Equipo 2	No Especificada
RNF03	Escalabilidad: El sistema debe ser capaz de adaptarse a un crecimiento en la cantidad de clientes y productos que maneja la farmacia.	NO	1.0	Equipo 2	No Especificada
RNF04	Disponibilidad: El sistema debe estar disponible en todo momento para su uso, con un tiempo de inactividad mínimo.	NO	1.0	Equipo 2	No Especificada
RNF05	Seguridad: El sistema debe contar con medidas de seguridad para proteger la información de accesos no autorizados.	NO	1.0	Equipo 2	No Especificada

IV. BUGS ENCONTRADOS

1. No existe una base de datos como tal, se realiza la creación en el archivo create_db.py

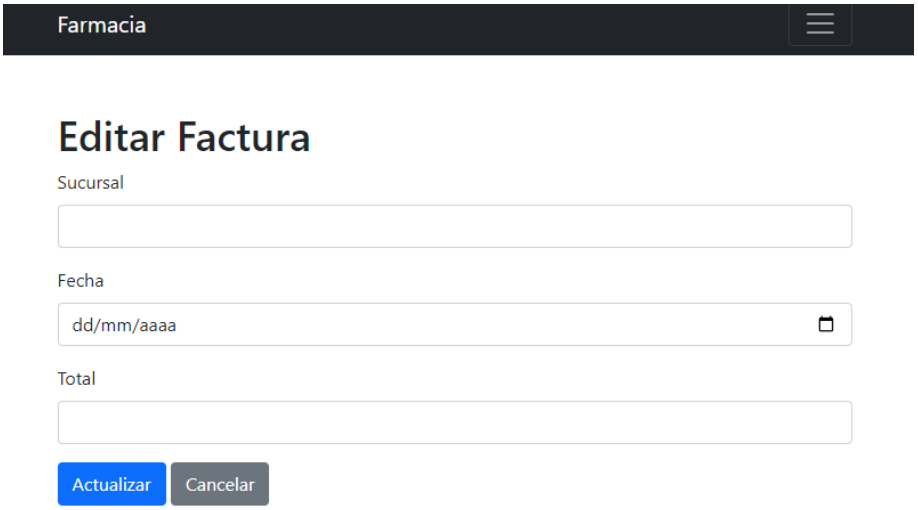
create_db.py > ...

```python
import sqlite3

# Crear una conexión a la base de datos
conn = sqlite3.connect('farmacia.db')

# Crear una tabla 'clientes' en la base de datos
conn.execute('''
    CREATE TABLE IF NOT EXISTS clientes (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        nombre TEXT NOT NULL,
        direccion TEXT NOT NULL,
        telefono TEXT NOT NULL,
        correo TEXT NOT NULL
    )
''')

# Crear una tabla 'productos' en la base de datos
conn.execute('''
    CREATE TABLE IF NOT EXISTS productos (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        nombre TEXT NOT NULL,
        precio REAL NOT NULL,
        stock INTEGER NOT NULL
    )
''')

# Crear una tabla 'facturas' en la base de datos
conn.execute('''
    CREATE TABLE IF NOT EXISTS facturas (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        cliente_id INTEGER NOT NULL,
        sucursal TEXT NOT NULL,
        fecha TEXT NOT NULL,
        total REAL NOT NULL,
        FOREIGN KEY (cliente_id) REFERENCES clientes (id)
    )
''')

# Crear una tabla 'detalle_factura' en la base de datos
```

```
create_db.py > ...
            factura_id INTEGER NOT NULL,
            producto_id INTEGER NOT NULL,
            cantidad INTEGER NOT NULL,
            subtotal REAL NOT NULL,
            FOREIGN KEY (factura_id) REFERENCES facturas (id),
            FOREIGN KEY (producto_id) REFERENCES productos (id)
        )
    ''')

    # Insertar datos de ejemplo en la tabla 'clientes'
    clientes = [
        ('Juan Pérez', 'Calle 123', '123456789', 'juanperez@gmail.com'),
        ('María López', 'Avenida 456', '987654321', 'marialopez@gmail.com'),
        ('Pedro Rodríguez', 'Calle 789', '456789123', 'pedrorodriguez@gmail.c
        ('Ana Gómez', 'Avenida 789', '321654987', 'anagomez@gmail.com'),
        ('Luisa García', 'Calle 456', '654987321', 'luisagarcia@gmail.com')
    ]

    conn.executemany('INSERT INTO clientes (nombre, direccion, telefono, corr

    # Insertar datos de ejemplo en la tabla 'productos'
    productos = [
        ('Paracetamol', 5.99, 100),
        ('Ibuprofeno', 7.5, 50),
        ('Amoxicilina', 12.99, 20),
        ('Vitamina C', 9.75, 80),
        ('Aspirina', 3.99, 60)
    ]

    conn.executemany('INSERT INTO productos (nombre, precio, stock) VALUES (?

    # Guardar los cambios
    conn.commit()

    # Cerrar la conexión a la base de datos
    conn.close()
```

2. La factura se puede editar, sin embargo esta no se debe de editar

Id	Cliente	Sucursal	Fecha	Total	Acciones
3	3	AQP	2023-05-11	21.48	Descargar Editar Eliminar

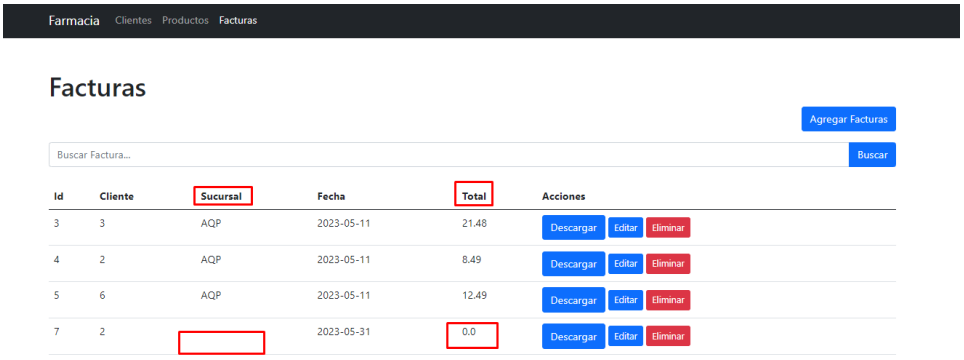


3. Al realizar archivo PDF de la factura, no aparecen los productos que se están comprando

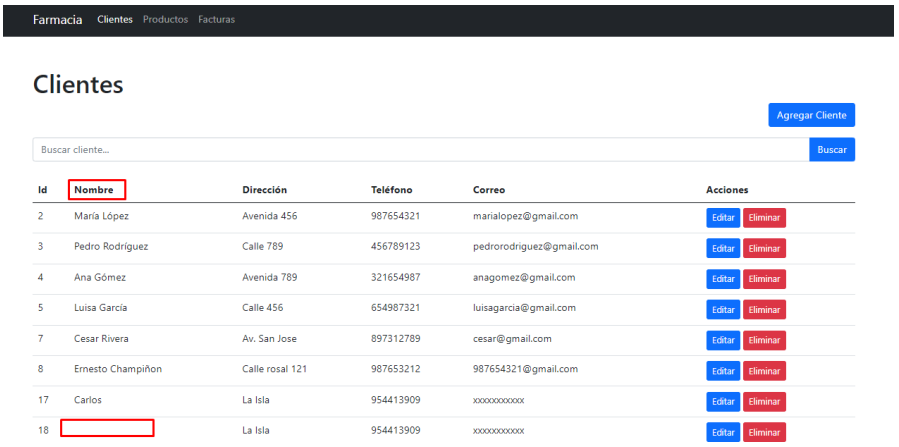
4. Los componentes en html no poseen de un identificador

```
<h1 class="mt-5">Agregar Factura</h1>
<form action="/agregar_factura" method="POST">
    <div class="mb-3">
        <label for="cliente" class="form-label">Cliente</label>
        <select class="form-select" id="cliente" name="cliente" required>
            {% for cliente in clientes %}
            <option value="{{ cliente[0] }}">{{ cliente[1] }}</option>
            {% endfor %}
        </select>
    </div>
```

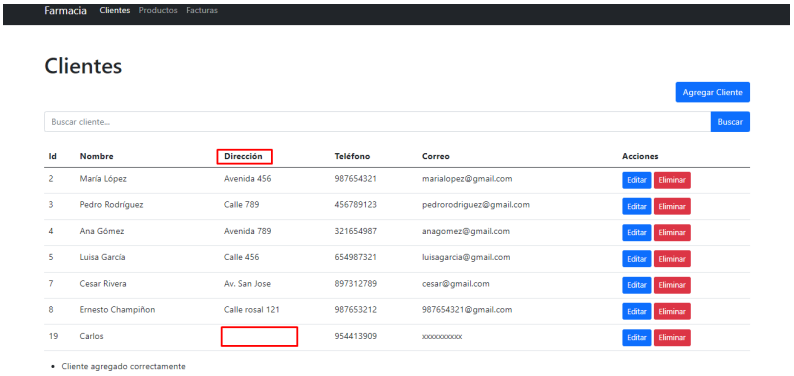
5. BUG 01: En la factura los campos de sucursal y productos pueden encontrarse sin completar y aun asi se emite la factura



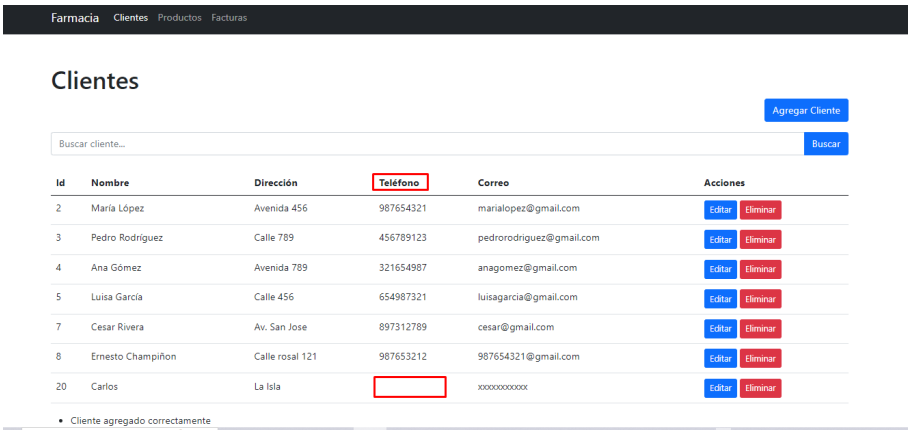
6. BUG 02: Al registrar un cliente el campo de nombre puede estar en blanco



7. BUG 03: Al registrar un cliente el campo de dirección puede estar en blanco



8. BUG 04: Al registrar un cliente el campo de teléfono puede estar en blanco



9. BUG 05: Al registrar un cliente el campo de correo puede estar en blanco

Farmacia Clientes Productos Facturas

Clientes

Agregar Cliente

Buscar cliente... Buscar

Id	Nombre	Dirección	Teléfono	Correo	Acciones
2	María López	Avenida 456	987654321	marialopez@gmail.com	Editar Eliminar
3	Pedro Rodríguez	Calle 789	456789123	pedrorodriguez@gmail.com	Editar Eliminar
4	Ana Gómez	Avenida 789	321654987	anagomez@gmail.com	Editar Eliminar
5	Luisa García	Calle 456	654987321	luisagarcia@gmail.com	Editar Eliminar
7	Cesar Rivera	Av. San Jose	897312789	cesar@gmail.com	Editar Eliminar
8	Ernesto Champiñon	Calle rosal 121	987653212	987654321@gmail.com	Editar Eliminar
21	Carlos	La Isla	954413909		Editar Eliminar

- Cliente agregado correctamente

10. BUG 06: En Productos el campo de nombre puede estar en blanco

Farmacia Clientes Productos Facturas

Productos

Agregar Producto

Buscar producto... Buscar

ID	Nombre	Precio	Stock	Acciones
1	Paracetamol	5.99	100	Editar Eliminar
2	Ibuprofeno	7.5	50	Editar Eliminar
3	Amoxicilina	12.99	20	Editar Eliminar
4	Vitamina C	9.75	80	Editar Eliminar
7	Miodel	2.5	25	Editar Eliminar
8	Gripa-C	2.0	10	Editar Eliminar
11	Z	-0.02	-1	Editar Eliminar
12		7.0	7	Editar Eliminar

11. BUG 07: Al ingresar un nuevo producto este puede tener un precio negativo

Productos

Agregar Producto

Buscar producto... Buscar

ID	Nombre	Precio	Stock	Acciones
1	Paracetamol	5.99	100	Editar Eliminar
2	Ibuprofeno	7.5	50	Editar Eliminar
3	Amoxicilina	12.99	20	Editar Eliminar
4	Vitamina C	9.75	80	Editar Eliminar
7	Miodel	2.5	25	Editar Eliminar
8	Gripa-C	2.0	10	Editar Eliminar
11	Z	-0.02	-1	Editar Eliminar

12. BUG 08: Al ingresar un nuevo producto este puede tener un stock negativo

Farmacia Clientes Productos Facturas

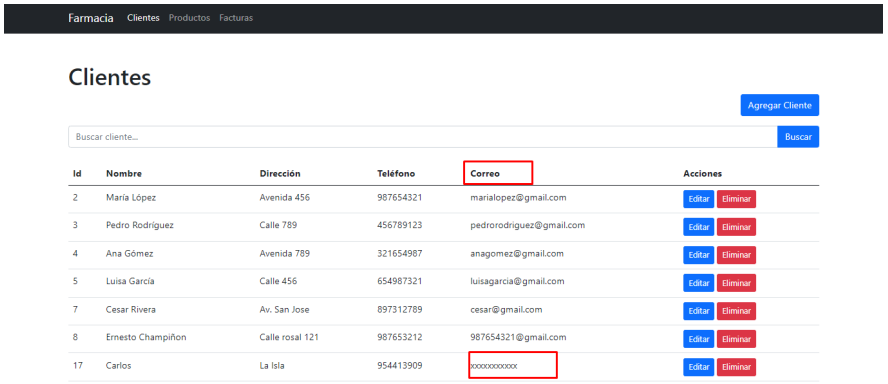
Productos

Agregar Producto

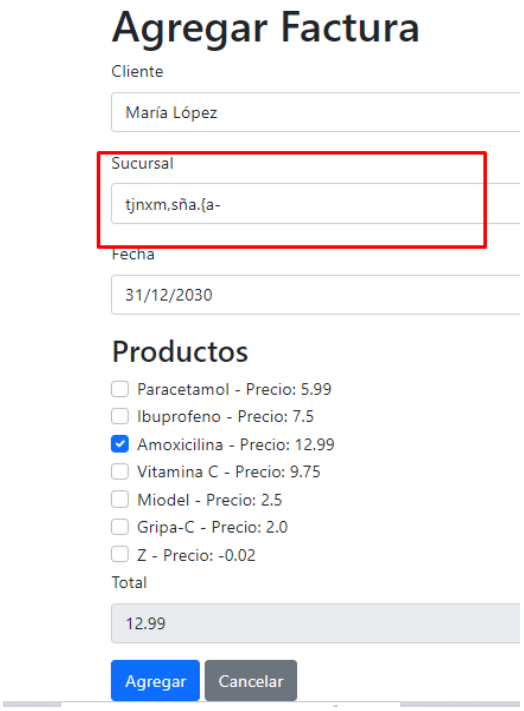
Buscar producto... Buscar

ID	Nombre	Precio	Stock	Acciones
1	Paracetamol	5.99	100	Editar Eliminar
2	Ibuprofeno	7.5	50	Editar Eliminar
3	Amoxicilina	12.99	20	Editar Eliminar
4	Vitamina C	9.75	80	Editar Eliminar
7	Miodel	2.5	25	Editar Eliminar
8	Gripa-C	2.0	10	Editar Eliminar
11	Z	-0.02	-1	Editar Eliminar

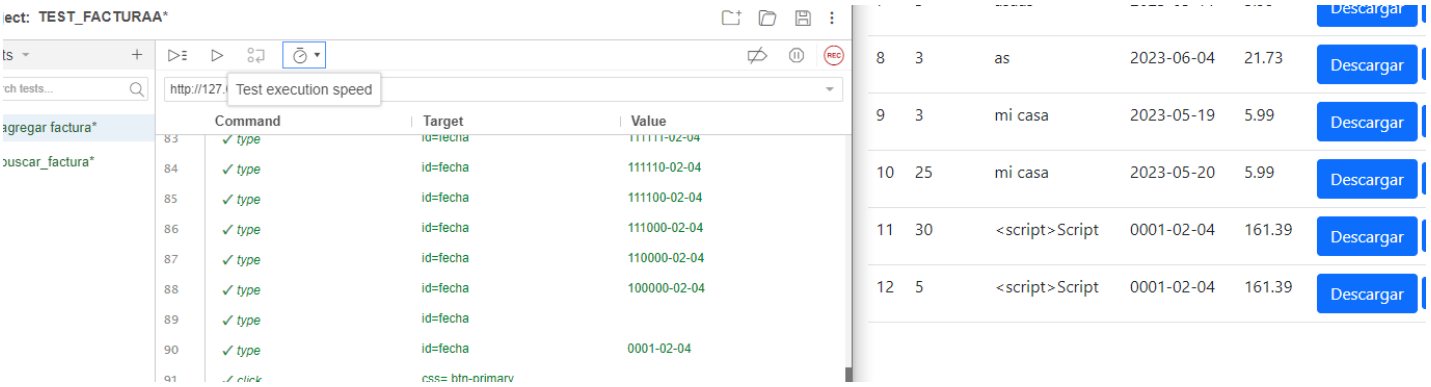
13. BUG 09: Al ingresar un cliente en el campo de correo acepta cualquier tipo de datos sin necesidad de un dominio

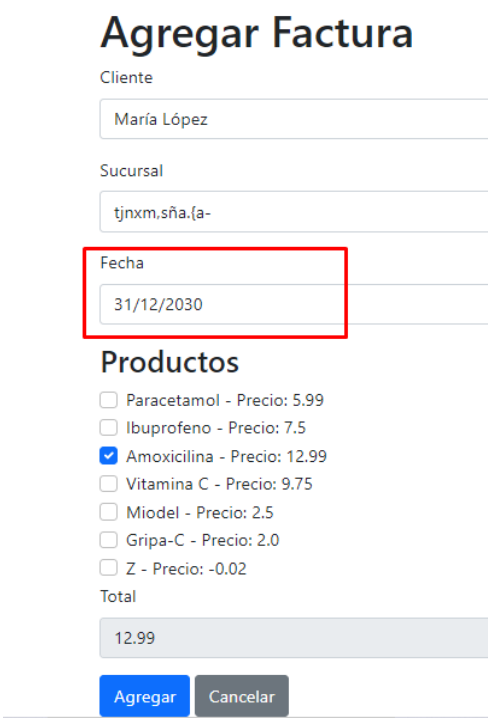


14. BUG 10: Al Ingresar a Factura al campo de la sucursales este no está vinculada a su base de datos

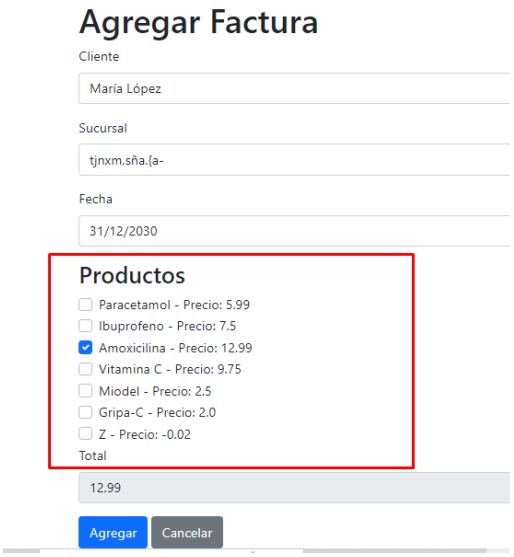


15. BUG 11: Al ingresar a Factura al momento de seleccionar la fecha, se puede seleccionar una fecha futura o pasada siendo esto un acto fraudulento, ademas, de igual forma acepta caracteres como <script> siendo aqui mucho mas grave siendo que se trata con la informacion de compras de los clientes y no existe ningun tipo de seguridad. algo bueno es que la fecha no acepta datos negativos ni que exceda la cantidad de dias o meses.

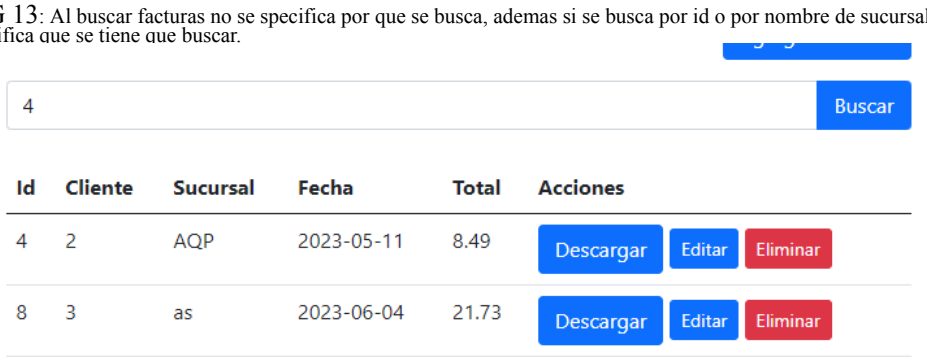


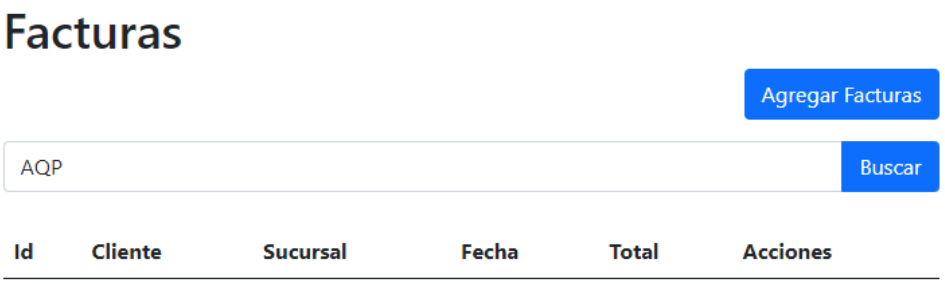


16. BUG 12: Al Ingresar Factura solo se considera el precio unitario por producto, no se puede incrementar el número de productos a comprar.

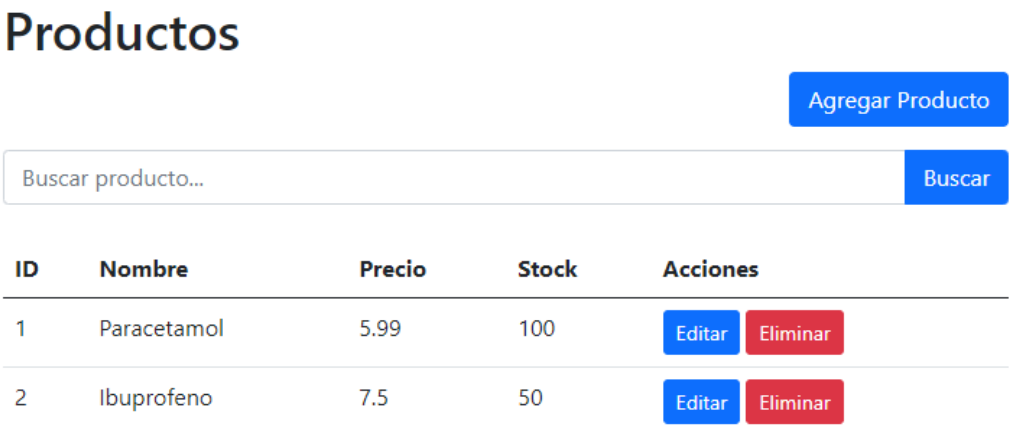


17. BUG 13: Al buscar facturas no se specifica por que se busca, ademas si se busca por id o por nombre de sucursal no da datos correctos, no se especifica que se tiene que buscar.





18. BUG 14: Al realizar la compra de productos el stock no se actualiza por cada compra.



19. Se repiten muchos productos dentro de la base de datos, no hay gestión de articulos por este lado, siendo un requisito funcional.

ID	Nombre	Precio	Stock	Acciones
1	Paracetamol	5.99	100	Editar Eliminar
2	Ibuprofeno	7.5	50	Editar Eliminar
3	Amoxicilina	12.99	20	Editar Eliminar
4	Vitamina C	9.75	80	Editar Eliminar
7	Miodel	2.5	25	Editar Eliminar
8	Gripa-C	2.0	10	Editar Eliminar
11	Paracetamol	5.99	100	Editar Eliminar
12	Ibuprofeno	7.5	50	Editar Eliminar
13	Amoxicilina	12.99	20	Editar Eliminar
14	Vitamina C	9.75	80	Editar Eliminar
15	Aspirina	3.99	60	Editar Eliminar
16	Paracetamol	5.99	100	Editar Eliminar
17	Ibuprofeno	7.5	50	Editar Eliminar
18	Amoxicilina	12.99	20	Editar Eliminar
19	Vitamina C	9.75	80	Editar Eliminar
20	Aspirina	3.99	60	Editar Eliminar

20. BUG 15: Al intentar testear el código se produjo un error dado que se intenta testear el llamado a HTML

test.py app.py

test.py > ...

```python
import sqlite3
import app

def test_obtener_clientes():
    clientes = app.obtener_clientes

    assert isinstance(clientes, list)

test_obtener_clientes()
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL Code

```
[Running] python -u "c:\Users\m3rct\Desktop\LOLO\FARMACIA3-main\test.py"
Traceback (most recent call last):
  File "c:\Users\m3rct\Desktop\LOLO\FARMACIA3-main\test.py", line 11, in <module>
    test_obtener_clientes()
  File "c:\Users\m3rct\Desktop\LOLO\FARMACIA3-main\test.py", line 7, in
  test_obtener_clientes
    assert isinstance(clientes, list)
AssertionError
```

Lín. 8, col. 5 Espacios: 4 UTF-8 CRLF Python 3.11.2 64-bit Port : 5500

```
nplates > <> agregar_cliente.html
1   {% extends "base.html" %}
2
3   {% block title %}Agregar Cliente{% endblock %}
4
5   {% block content %}
6       <h1 class="mt-5">Agregar Cliente</h1>
7       <form action="/agregar_cliente" method="POST" class="needs-validatic
8           <div class="mb-3">
9               <label for="nombre" class="form-label">Nombre</label>
10              <input type="text" class="form-control" id="nombre" name="nc
11              <div class="invalid-feedback">Solo se permiten letras.</div>
12          </div>
13          <div class="mb-3">
14              <label for="direccion" class="form-label">Dirección</label>
15              <input type="text" class="form-control" id="direccion" name=
16          </div>
17          <div class="mb-3">
18              <label for="telefono" class="form-label">Teléfono</label>
19              <input type="text" class="form-control" id="telefono" name="
20              <div class="invalid-feedback">Solo se permiten 9 números.</d
21          </div>
22          <div class="mb-3">
23              <label for="correo" class="form-label">Correo</label>
24              <input type="email" class="form-control" id="correo" name="c
25              <div class="invalid-feedback">Ingrese un correo electrónico
26          </div>
27          <button type="submit" class="btn btn-primary">Agregar</button>
28          <a href="/clientes" class="btn btn-secondary">Cancelar</a>
29      </form>
30
31      <script>
32          // Validación adicional en el lado del cliente usando JavaScript
33          document.getElementById('nombre').addEventListener('input', func
34              this.value = this.value.replace(/[^A-Za-z ]/g, '');
35          });
36
37          document.getElementById('telefono').addEventListener('input', fu
38              this.value = this.value.replace(/[^0-9]/g, '');
```

21. Como el test no funcionó se agregó un id al elemento para poder hacer el test a los casos y de igual manera no se ejecutaban de manera correcta.

```
test.py 3, U ●   index.html M
test > test.py > ...
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time
options = webdriver.ChromeOptions()
options.add_argument('--start-maximized')
options.add_argument('--disable-extensions')

driver = webdriver.Chrome("chromedriver.exe")
driver.get('http://127.0.0.1:5000/')
btnFactura = driver.find_element(By.ID,'btnFact')
time.sleep(100)
driver.quit()
```

```
<div class="row">
    <div class="col-md-4">
        <div class="card rounded shadow-sm">
            <div class="card-body">
                <h2 class="card-title"><i class="fas fa-file-invoice"></i> Facturas</h2>
                <p class="card-text">Accede a la gestión de facturas para crear, buscar y ver detalles de las facturas.</p>
                <a href="/facturas" id ='btnFact' class="btn btn-primary">Gestionar Facturas</a>
```

```
PS D:\UCSM\Testing Practica\Equipo enemigo\FARMACIA3\test> python test.py
D:\UCSM\Testing Practica\Equipo enemigo\FARMACIA3\test\test.py:9: DeprecationWarning: executable_path has been deprecated, please pass in a Service object
  driver = webdriver.Chrome("chromedriver.exe")

DevTools listening on ws://127.0.0.1:64806/devtools/browser/75c662aa-74a4-49ea-bc2d-8fea7abc4591
Traceback (most recent call last):
  File "D:\UCSM\Testing Practica\Equipo enemigo\FARMACIA3\test\test.py", line 9, in <module>
    driver = webdriver.Chrome("chromedriver.exe")
  File "C:\Users\USER\AppData\Local\Programs\Python\Python310\lib\site-packages\selenium\webdriver\chrome\webdriver.py", line 84, in __init__
    super().__init__(
  File "C:\Users\USER\AppData\Local\Programs\Python\Python310\lib\site-packages\selenium\webdriver\chromium\webdriver.py", line 104, in __init__
    super().__init__(
  File "C:\Users\USER\AppData\Local\Programs\Python\Python310\lib\site-packages\selenium\webdriver\remote\webdriver.py", line 286, in __init__
    self.start_session(capabilities, browser_profile)
  File "C:\Users\USER\AppData\Local\Programs\Python\Python310\lib\site-packages\selenium\webdriver\remote\webdriver.py", line 378, in start_session
    response = self.execute(Command.NEW_SESSION, parameters)
  File "C:\Users\USER\AppData\Local\Programs\Python\Python310\lib\site-packages\selenium\webdriver\remote\webdriver.py", line 440, in execute
    self.error_handler.check_response(response)
  File "C:\Users\USER\AppData\Local\Programs\Python\Python310\lib\site-packages\selenium\webdriver\remote\errorhandler.py", line 245, in check_response
```

V. CONCLUSIONES

- La base de datos se crea al ejecutar el código, por lo cual no es la mejor opción ya que al no estar conectada a internet, no se puede conectar de forma remota.
- Con el test de caja negra se logró la validación de los requisitos funcionales y no funcionales, además de evaluar la usabilidad del sistema.
- El sistema de facturación presenta varios problemas de seguridad, empezando con que no existe un login que permita restringir o permitir el ingreso, el uso de los caracteres “<”, “>” haciendo que la seguridad sea mucho más afectada.
- El documento que se emite en la factura no tiene los datos que debería tener una factura, faltando el nombre del cliente, nombre del producto, cantidad entre otros.
- En conclusión el sistema presenta diversos errores, cabe resaltar que el sistema tiene la mayor parte del código en HTML, por lo cual solo se pudo realizar testeo de caja negra.