

Asignación 3 - Validación y Corrección de Código Inseguro

Parte A — Detección (8 pts)

1. Lee el código y responde:

- a) ¿Por qué es inseguro aceptar cualquier texto sin validación?

Porque esto le permite a un atacante introducir código o datos maliciosos que la aplicación procesará y potencialmente ejecutará sin control. También puede permitir accesos no autorizados o la manipulación de los datos del sistema, afectando la integridad y privacidad.

- b) ¿Qué tipo de ataque podría aprovechar esto?

El ataque critico que se beneficia de esta vulnerabilidad es la inyección de código (Code injection). Donde un atacante inserta comandos o código malicioso por una entrada que la aplicación no valido antes de procesarla.

2. Ejecuta el programa y prueba introducir:

- Texto normal (ejemplo: *Hola mundo*).

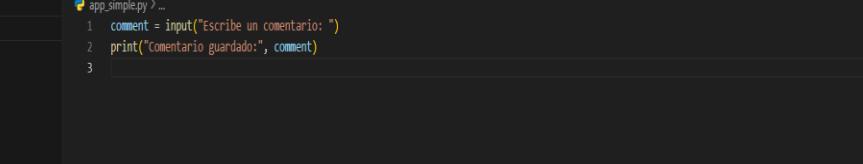
The screenshot shows the Microsoft Visual Studio Code interface. On the left, the Explorer sidebar shows a folder named 'ASIGNACIÓN3' containing 'app_simple.py'. The code editor window displays the following Python script:

```
comment = input("Escribe un comentario: ")
print("Comentario guardado:", comment)
```

Below the code editor is the Terminal panel, which shows the output of running the script:

```
PS C:\Users\bm68\OneDrive\Desktop\Asignación3> & C:/Users/bm68/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/bm68/OneDrive/Desktop/Asignación3/app_simple.py
Escribe un comentario: Hola Mundo!
Comentario guardado: Hola Mundo!
```

- Texto con etiquetas HTML o comandos sospechosos (ejemplo: <script>)



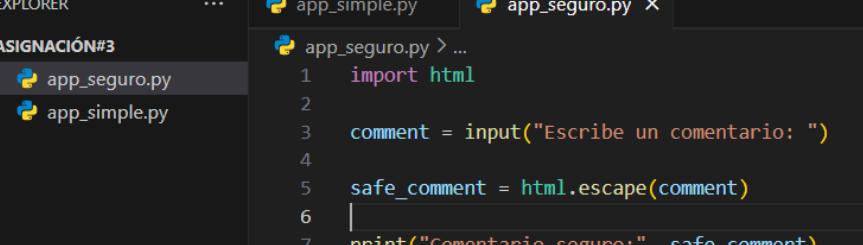
The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The top bar includes the standard File, Edit, Selection, View, Go, Run, Terminal, Help menus, and a search bar labeled "Q Asignación#3". On the far right are window control buttons. The left sidebar has icons for Explorer, ASIGNACIÓN#3, OUTLINE, and TIMELINE. The main editor area displays a Python file named "app_simple.py" with the following code:

```
comment = input("Escribe un comentario: ")
print("Comentario guardado:", comment)
```

The terminal at the bottom shows the command PS C:\Users\bmm68\OneDrive\Desktop\Asignación#3 & C:/Users/bmm68/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/bmm68/OneDrive/Desktop/Asignación#3/app_simple.py being run. The output of the script is displayed as:

```
Escribe un comentario: <script>alert('Hackeado');</script>
Comentario guardado: <script>alert('Hackeado');</script>
```

Parte B — Corrección (10 pts)



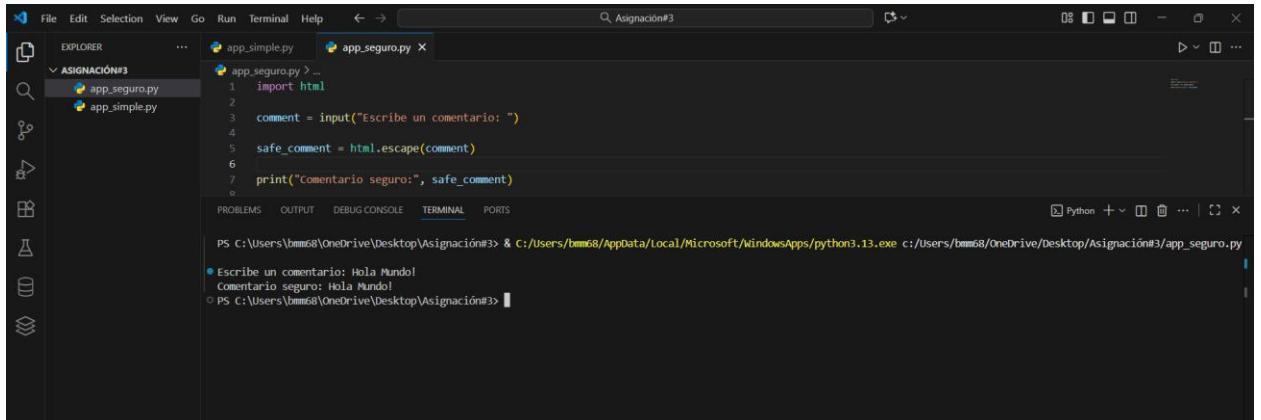
The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Terminal:** Shows the command `app_seguo.py`.
- Explorer:** Shows a folder named "ASIGNACIÓN#3" containing two files: `app_seguo.py` and `app_simple.py`. The `app_seguo.py` file is currently open.
- Code Editor:** Displays the following Python code:

```
1 import html
2
3 comment = input("Escribe un comentario: ")
4
5 safe_comment = html.escape(comment)
6
7 print("Comentario seguro:", safe_comment)
8
```

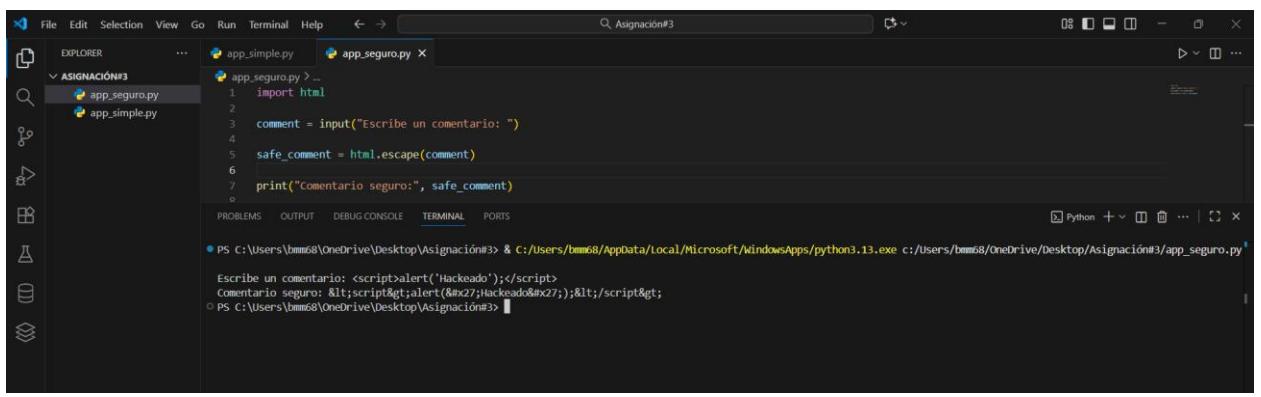
- En Python la función de `html.escape()` es la de convertir caracteres especiales de una cadena de texto hacia equivalentes en entidades HTML. Tiene como propósito prevenir vulnerabilidades de seguridad y asegurar que el texto se muestre de manera correcta en un navegador web.

Parte C — Verificación (5 pts)



```
File Edit Selection View Go Run Terminal Help ⏮ ⏯ Asignación#3
EXPLORER ASIGNACIÓN#3
app_seguo.py app_seguo.py
app_seguo.py > ...
1 import html
2
3 comment = input("Escribe un comentario: ")
4
5 safe_comment = html.escape(comment)
6
7 print("Comentario seguro:", safe_comment)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\bmm68\OneDrive\Desktop\Asignación#3> & C:/Users/bmm68/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/bmm68/OneDrive/Desktop/Asignación#3/app_seguo.py
● Escribe un comentario: Hola Mundo!
Comentario seguro: Hola Mundo!
○ PS C:\Users\bmm68\OneDrive\Desktop\Asignación#3>
```



```
File Edit Selection View Go Run Terminal Help ⏮ ⏯ Asignación#3
EXPLORER ASIGNACIÓN#3
app_seguo.py app_seguo.py
app_seguo.py > ...
1 import html
2
3 comment = input("Escribe un comentario: ")
4
5 safe_comment = html.escape(comment)
6
7 print("Comentario seguro:", safe_comment)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\bmm68\OneDrive\Desktop\Asignación#3> & C:/Users/bmm68/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/bmm68/OneDrive/Desktop/Asignación#3/app_seguo.py
● Escribe un comentario: <script>alert('Hackeado');</script>
Comentario seguro: &lt;script&gt;alert ('Hackeado');&lt;/script&gt;
○ PS C:\Users\bmm68\OneDrive\Desktop\Asignación#3>
```

Preguntas Teóricas (2 pts)

1. ¿Qué principio de la seguridad del software se está aplicando al validar la entrada del usuario?

El principio que se esta utilizando es el de la validación robusta de entradas. En donde el programa limpia y verifica los datos que se están entrando para evitar datos maliciosos antes de procesarlo.

2. ¿Qué impacto tendría no sanitizar los datos en la Confidencialidad y la Integridad del sistema?

El impacto en la Confidencialidad se podría notar por parte que podrían ocurrir fugas de datos, permitiendo la extracción o monitoreo de información personal, credenciales y datos almacenados. Cuanto a la Integridad del sistema se puede ver afectado por atacantes que pueden injectar código que altere el funcionamiento normal de la aplicación, manipule registros o bases de datos.