

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?

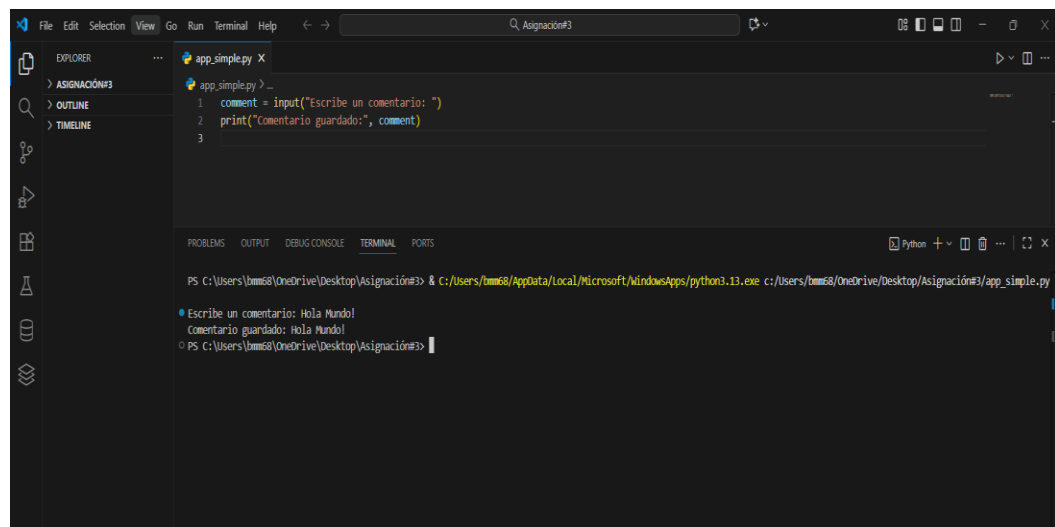
Porque esto le permite a un atacante introducir código o datos maliciosos que la aplicación procesará y potencialmente ejecutará sin control. También puede permitir accesos no autorizados o la manipulación de los datos del sistema, afectando la integridad y privacidad.

b) ¿Qué tipo de ataque podría aprovechar esto?

El ataque crítico 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 valida antes de procesarla.

2. Ejecuta el programa y prueba introducir:

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



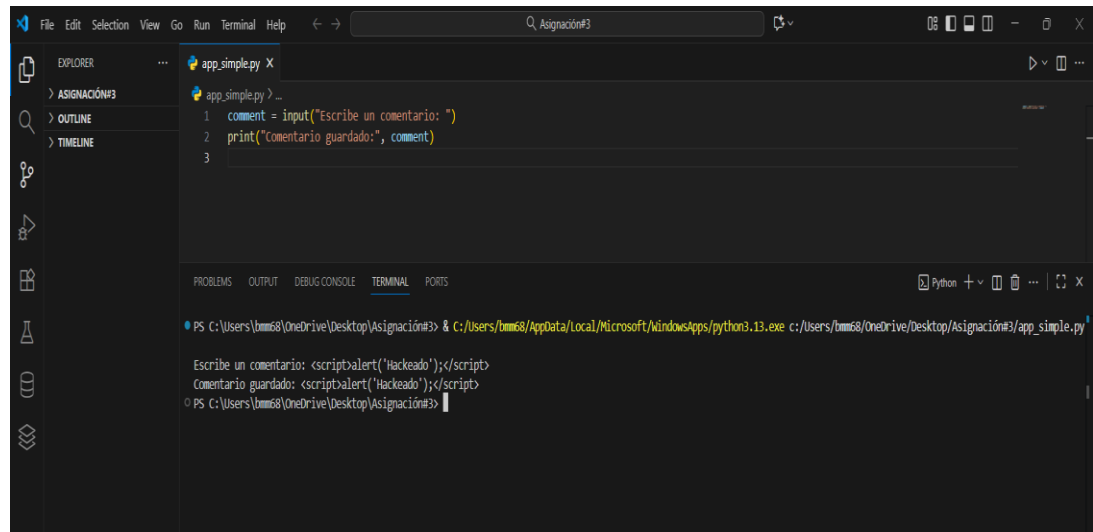
The screenshot shows a Visual Studio Code window with a file named `app_simple.py` open. The code in the editor is:

```
1 comment = input("Escribe un comentario: ")
2 print("Comentario guardado:", comment)
3
```

Below the editor, the TERMINAL panel is active, showing the command to run the script and its output:

```
PS C:\Users\lmm68\OneDrive\Desktop\Asignación3> & C:/Users/lmm68/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/lmm68/OneDrive/Desktop/Asignación3/app_simple.py
• Escribe un comentario: Hola Mundo!
Comentario guardado: Hola Mundo!
PS C:\Users\lmm68\OneDrive\Desktop\Asignación3>
```

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



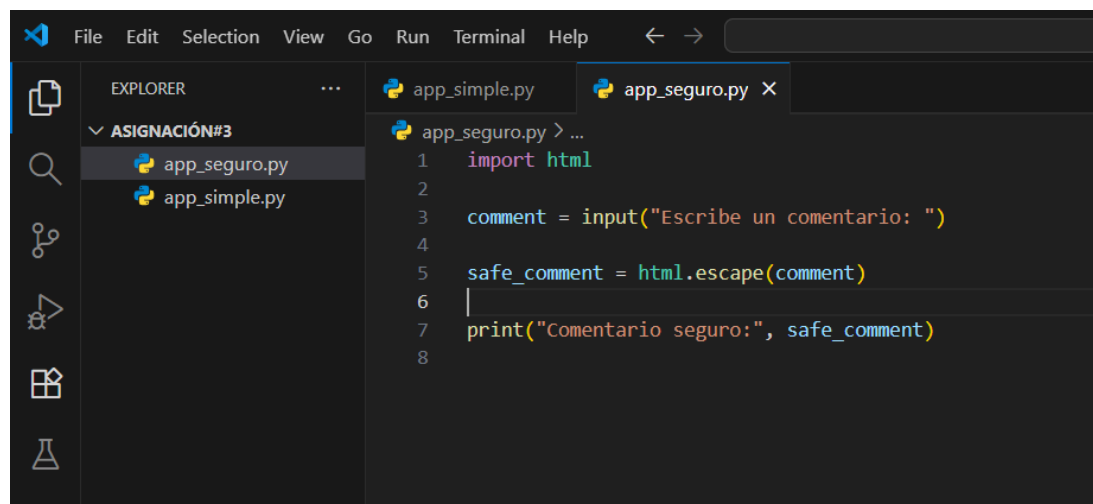
The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows a project named 'ASIGNACIÓN#3' containing two files: 'app_seguro.py' and 'app_simple.py'. The main editor displays 'app_simple.py' with the following code:

```
1 comment = input("Escribe un comentario: ")
2 print("Comentario guardado:", comment)
3
```

The integrated terminal at the bottom shows the command to run the script and its output:

```
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
Escribe un comentario: <script>alert('Hackeado');</script>
Comentario guardado: <script>alert('Hackeado');</script>
PS C:\Users\bmm68\OneDrive\Desktop\Asignación#3>
```

Parte B — Corrección (10 pts)

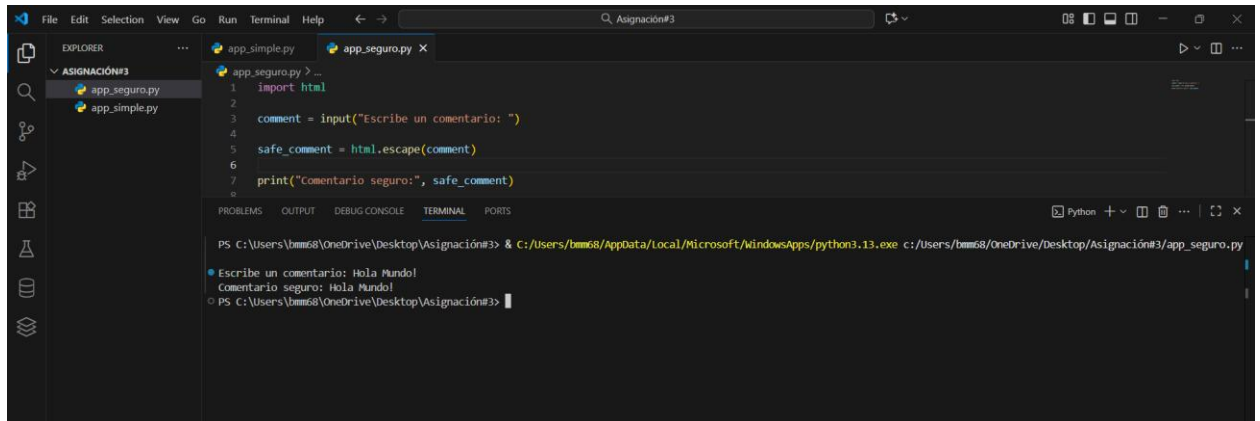


The screenshot shows the Visual Studio Code interface with two files open: 'app_simple.py' and 'app_seguro.py'. The main editor displays 'app_seguro.py' with the following code:

```
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)

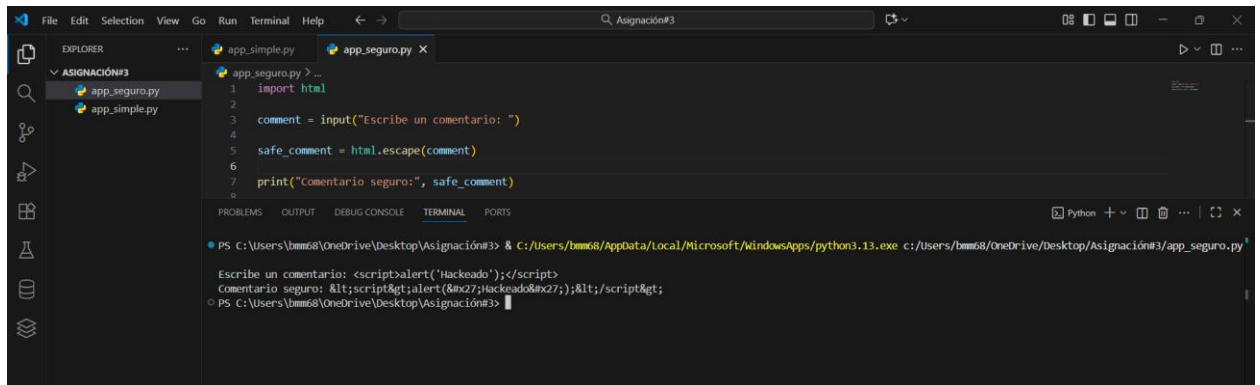


The screenshot shows a Visual Studio Code editor with a file explorer on the left containing 'app_seguro.py' and 'app_simple.py'. The main editor displays 'app_seguro.py' with the following code:

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

The terminal at the bottom shows the command to run the script and its output:

```
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_seguro.py
Escribe un comentario: Hola Mundo!
Comentario seguro: Hola Mundo!
PS C:\Users\bmm68\OneDrive\Desktop\Asignación#3>
```



The screenshot shows the same Visual Studio Code editor with the same Python script. The terminal shows the command to run the script and its output for a malicious input:

```
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_seguro.py
Escribe un comentario: <script>alert("Hackeado");</script>
Comentario seguro: &lt;script&gt;alert(&#x27;hackeado&#x27;);&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 está 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 inyectar código que altere el funcionamiento normal de la aplicación, manipule registros o bases de datos.