

[Respuesta_Corta]

En el método GET, los datos del formulario HTML se envían en la URL después del carácter '?'. Esa cadena se denomina QUERY_STRING, estando constituida por pares de "name=value" separados ¿por qué carácter?.

[Respuesta_Corta]

En el método GET, los datos del formulario HTML se envían en la URL después del carácter '?'. Esa cadena se denomina QUERY_STRING, estando constituida por pares de "name=value" separados ¿por qué carácter?.

Solución:	&
------------------	--------------

[Test]

Suponga que desde un navegador se envía a un servidor Web un mensaje HTTP con el método POST, que incluye los datos de un formulario HTML. Si dicho mensaje solicita un programa CGI, éste recibe la información de los campos del formulario mediante:

- (a) La entrada estándar.
- (b) La variable de entorno **DATA_STRING**.
- (c) La variable de entorno **QUERY_STRING**.

[Test]

Suponga que desde un navegador se envía a un servidor Web un mensaje HTTP con el método POST, que incluye los datos de un formulario HTML. Si dicho mensaje solicita un programa CGI, éste recibe la información de los campos del formulario mediante:

- ✓ (a) La entrada estándar.
- (b) La variable de entorno **DATA_STRING**.
- (c) La variable de entorno **QUERY_STRING**.

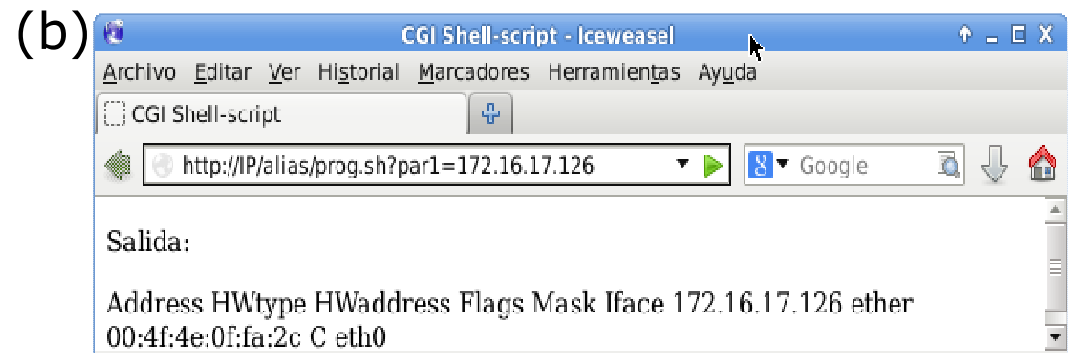
[Respuesta_Corta]

Indique qué captura de pantalla puede corresponder a la salida de un navegador Web tras solicitar a un servidor la URL:

`http://IP/alias/prog.sh?172.16.17.126`

```
#!/bin/sh
echo "Content-type: text/html"
echo
echo "<!DOCTYPE html>"
echo "<html><head><title>"
echo "CGI Shell-script</title></head><body>"
echo "<p>Salida:</p>"
arp -n ${QUERY_STRING}
echo "</body></html>"
```

prog.sh



(c) Ninguna, ese script no cumple la interfaz CGI.

```

<!DOCTYPE html>
<html><head><title>
CGI Shell-script</title></head><body>
<p>Salida:</p>
Address          HWtype  HWaddress      Flags Mask    Iface
172.16.17.126    ether   00:4f:4e:0f:fa:2c  C           eth0
</body></html>

```

HTTP Response <http://IP/alias/prog.sh?172.16.17.126>

Cab. {

```

#!/bin/sh
echo "Content-type: text/html"
echo
echo "<!DOCTYPE html>"
echo "<html><head><title>"
echo "CGI Shell-script</title></head><body>"
echo "<p>Salida:</p>"
arp -n ${QUERY_STRING}
echo "</body></html>"

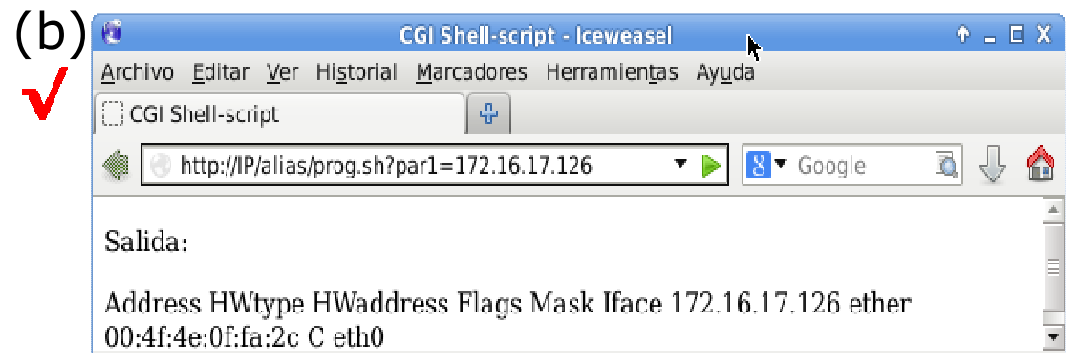
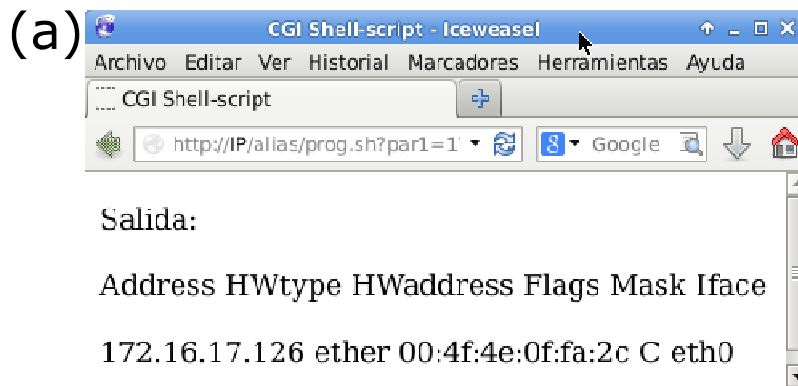
```

Cuerpo {

← Línea en blanco

prog.sh

Cuerpo de
HTTP
Response



(c) Ninguna, ese script no cumple la interfaz CGI.

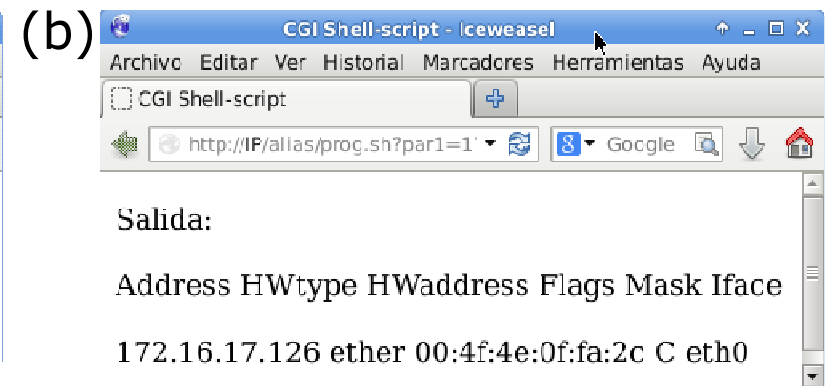
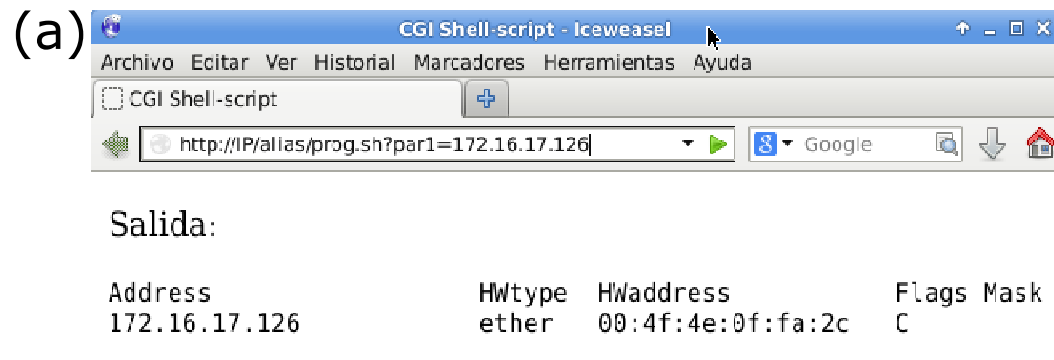
[Respuesta_Corta]

Indique qué captura de pantalla puede corresponder a la salida de un navegador Web tras solicitar a un servidor la URL:

`http://IP/alias/prog.sh?172.16.17.126`

```
#!/bin/sh
echo "Content-type: text/html"
echo
echo "<!DOCTYPE html>"
echo "<html><head><title>"
echo "CGI Shell-script</title></head><body>"
echo "<p>Salida:</p><pre>"
arp -n ${QUERY_STRING}
echo "</pre></body></html>"
```

prog.sh



(c) Ninguna, ese script no cumple la interfaz CGI.

```

<!DOCTYPE html>
<html><head><title>
CGI Shell-script</title></head><body>
<p>Salida:</p><pre>
Address          HWtype  HWaddress      Flags Mask    Iface
172.16.17.126    ether    00:4f:4e:0f:fa:2c  C             eth0
</pre></body></html>

```

HTTP Response <http://IP/alias/prog.sh?172.16.17.126>

Cab. {

```

#!/bin/sh
echo "Content-type: text/html"
echo
echo "<!DOCTYPE html>"
echo "<html><head><title>"
echo "CGI Shell-script</title></head><body>"
echo "<p>Salida:</p><pre>"
arp -n ${QUERY_STRING}
echo "</pre></body></html>"

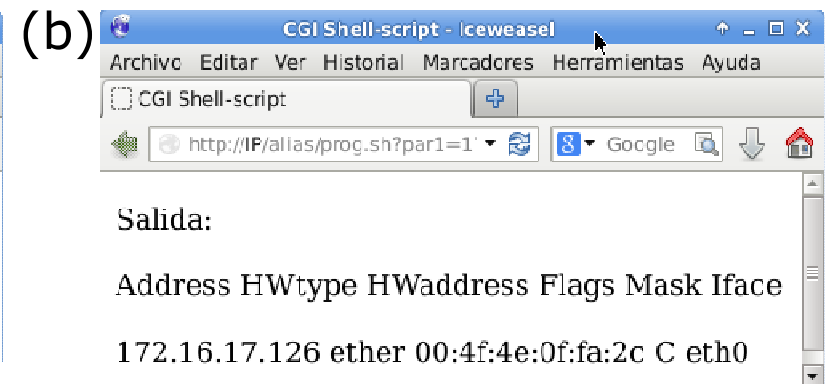
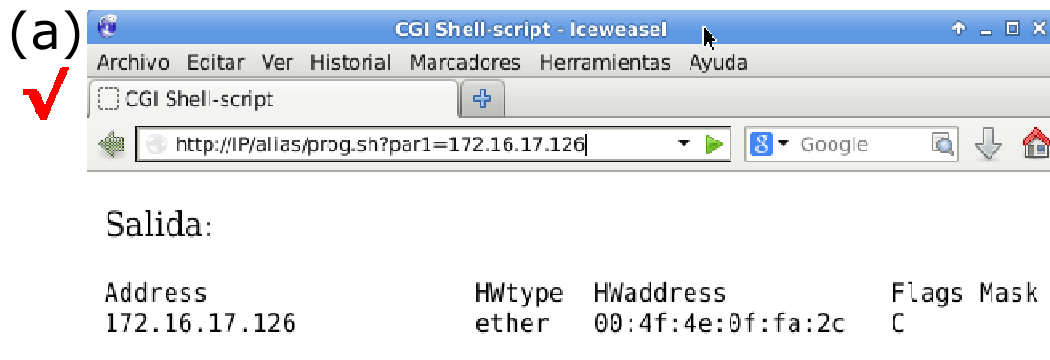
```

Cuerpo {

← Línea en blanco

prog.sh

Cuerpo de
HTTP
Response



(c) Ninguna, ese script no cumple la interfaz CGI.

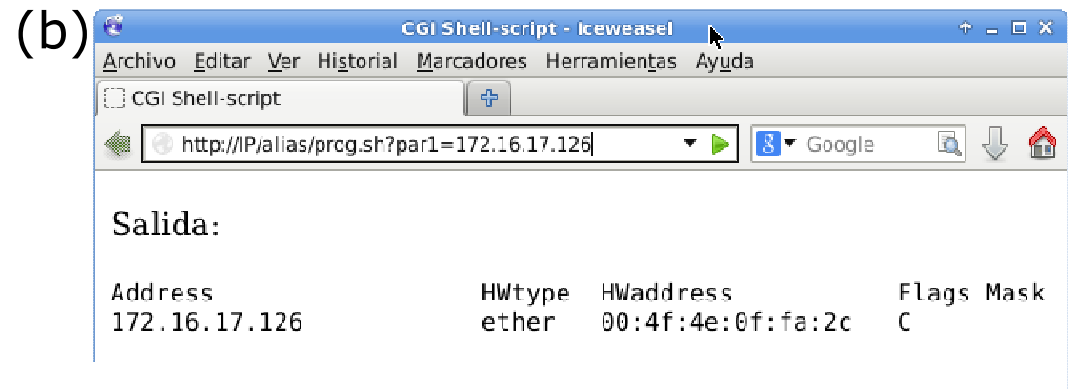
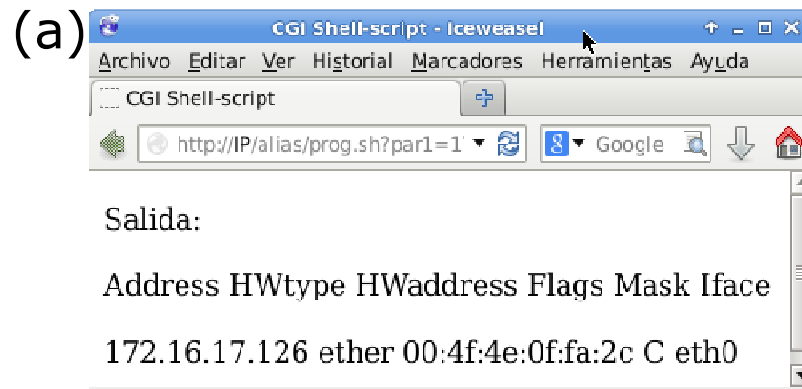
[Respuesta_Corta]

Indique qué captura de pantalla puede corresponder a la salida de un navegador Web tras solicitar a un servidor la URL:

`http://IP/alias/prog.sh?172.16.17.126`

```
#!/bin/sh
echo "Content-type: text/html"
echo "<!DOCTYPE html>"
echo "<html><head><title>"
echo "CGI Shell-script</title></head><body>"
echo "<p>Salida:</p><pre>"
arp -n ${QUERY_STRING}
echo "</pre></body></html>"
```

prog.sh



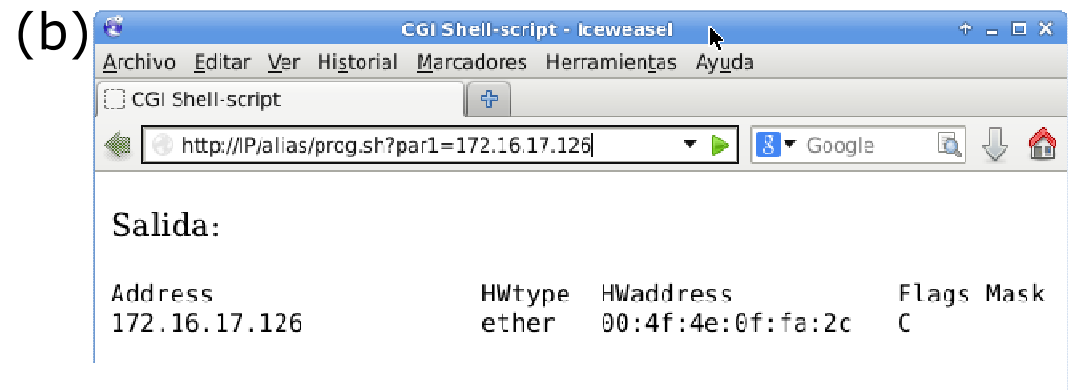
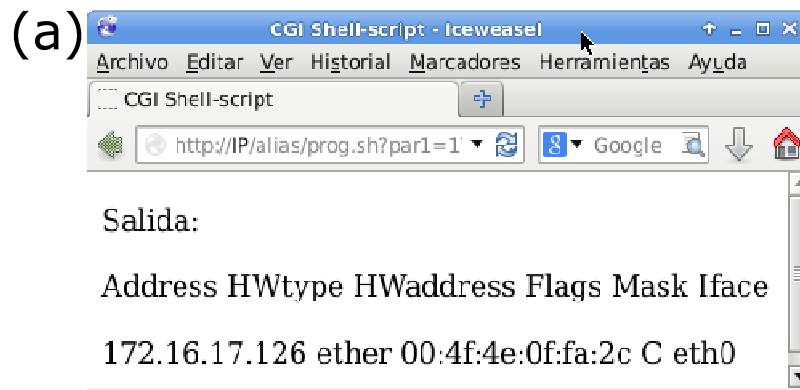
(c) Ninguna, ese script no cumple la interfaz CGI.

[Respuesta_Corta]

Indique qué captura de pantalla puede corresponder a la salida de un navegador Web tras solicitar a un servidor la URL:

HTTP `http://IP/alias/prog.sh?172.16.17.126`
Response

Cab. { `#!/bin/sh` prog.sh
`echo "Content-type: text/html"` ← **FALTA** Línea en blanco
Cuerpo { `echo "<!DOCTYPE html>"`
`echo "<html><head><title>"`
`echo "CGI Shell-script</title></head><body>"`
`echo "<p>Salida:</p><pre>"`
`arp -n ${QUERY_STRING}`
`echo "</pre></body></html>"`



✓(c) Ninguna, ese script no cumple la interfaz CGI.

[TEST]

Dada la siguiente aplicación web:

```
<html><head>...</head><body><h1>Cabecera de la pagina</h1>
<%=
    String cadena="0";
    for( int i=1; i < 10; i++ )
        cadena = cadena +i;
    %>
<form action="servlets/servlet_file" method="post">
    Valor: <input type="text" name="value"><BR>
    <input type="Submit" name="Enviar">
</form>
</body></html>
```

Indique cual es la opción correcta:

- (a) Dicha aplicación web es un servlet.
- (b) Un servlet no puede ser invocado desde este tipo de páginas.
- (c) La etiqueta JSP que alberga el código Java que contiene el bucle "for" debería ser "<%" en lugar de "<%= ", puesto que ésta última corresponde a una expresión JSP.

[TEST]

Dada la siguiente aplicación web:

```
<html><head>...</head><body><h1>Cabecera de la pagina</h1>
<%=
    String cadena="0";
    for( int i=1; i < 10; i++ )
        cadena = cadena +i;
    %>
<form action="servlets/servlet_file" method="post">
    Valor: <input type="text" name="value"><BR>
    <input type="Submit" name="Enviar">
</form>
</body></html>
```

Indique cual es la opción correcta:

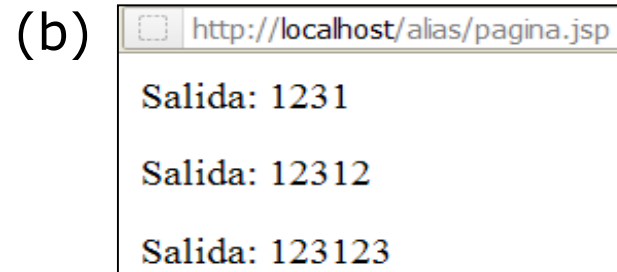
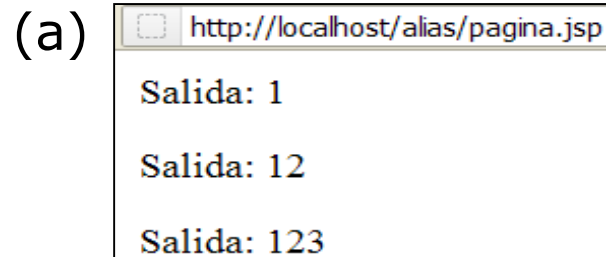
- (a) Dicha aplicación web es un servlet.
- (b) Un servlet no puede ser invocado desde este tipo de páginas.
- ✓ (c) La etiqueta JSP que alberga el código Java que contiene el bucle "for" debería ser "<%" en lugar de "<%= ", puesto que ésta última corresponde a una expresión JSP.

[Test]

Indique qué captura de pantalla puede corresponder a la salida de un navegador Web tras solicitar a un servidor el fichero "index.jsp" indicado, sabiendo que dicha página ha sido solicitada otra vez antes desde otro navegador web:

index.jsp

```
<html><body>
<% String cadena=""; %>
<%  for( int i=1; i < 4; i++ ) {
      cadena = cadena +i;
%>
      <p>Salida: <%= cadena %></p>
<% } %>
</body></html>
```



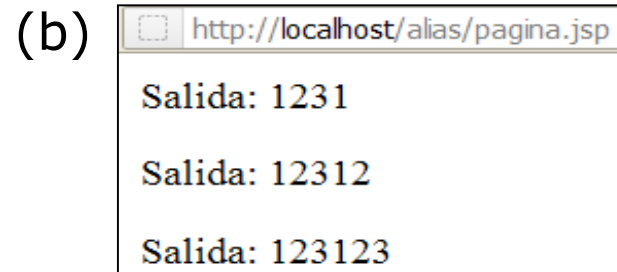
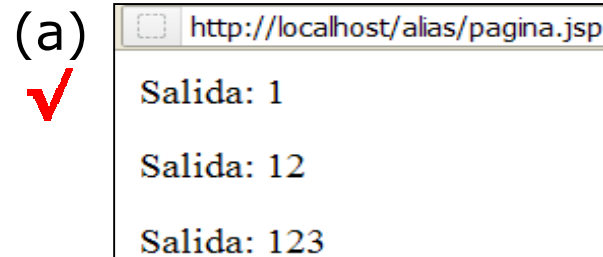
(c) Ninguna de las anteriores.

[Test]

Indique qué captura de pantalla puede corresponder a la salida de un navegador Web tras solicitar a un servidor el fichero "index.jsp" indicado, sabiendo que dicha página ha sido solicitada otra vez antes desde otro navegador web:

index.jsp

```
<html><body>
<% String cadena=""; %>
<%  for( int i=1; i < 4; i++ ) {
      cadena = cadena +i;
%>
      <p>Salida: <%= cadena %></p>
<% } %>
</body></html>
```



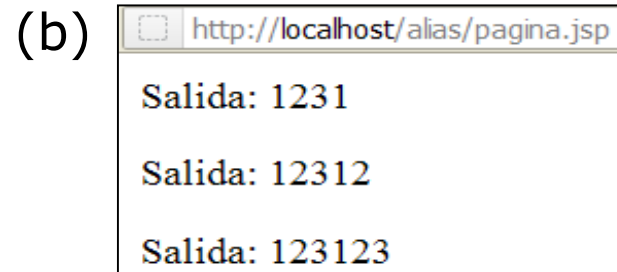
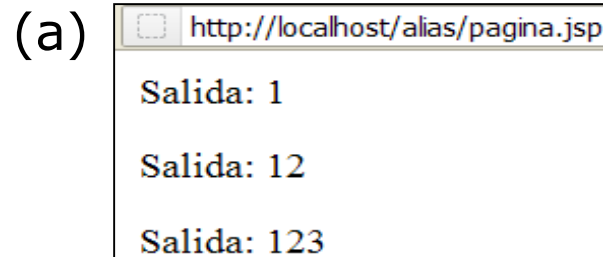
(c) Ninguna de las anteriores.

[Test]

Indique qué captura de pantalla puede corresponder a la salida de un navegador Web tras solicitar a un servidor el fichero "index.jsp" indicado, sabiendo que dicha página ha sido solicitada otra vez antes desde otro navegador web:

index.jsp

```
<html><body>
<%! String cadena=""; %>
<%  for( int i=1; i < 4; i++ ) {
      cadena = cadena +i;
%>
      <p>Salida: <%= cadena %></p>
<% } %>
</body></html>
```



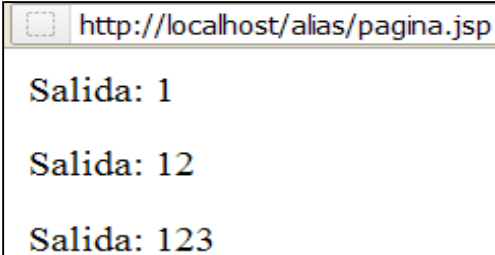
(c) Ninguna de las anteriores.

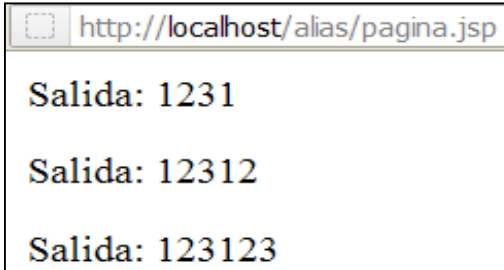
[Test]

Indique qué captura de pantalla puede corresponder a la salida de un navegador Web tras solicitar a un servidor el fichero "index.jsp" indicado, sabiendo que dicha página ha sido solicitada otra vez antes desde otro navegador web:

index.jsp

```
<html><body>
<%! String cadena=""; %>
<% for( int i=1; i < 4; i++ ) {
    cadena = cadena +i;
%>
    <p>Salida: <%= cadena %></p>
<% } %>
</body></html>
```

(a) 

(b) 

(c) Ninguna de las anteriores.

[TEST]

Una Aplicación Web en servidor con Java tiene los dos ficheros siguientes:

index.jsp	<pre>1 <%@ page language="java" import="java.util.*,java.io.*" session="true" %> 2 <%@ page import="paqueteJava.*" %> 3 <html><body>Pagina JSP: mensaje inicial<p> <% 4 clase parametros = new clase(); 5 request.getParameter("cadena"); parametros.funcion(response); 6 System.out.println("Hola"); %> </p></html></pre>
clase.java	<pre>package paqueteJava; import java.util.*; import java.io.*; import javax.servlet.*; import javax.servlet.jsp.*; public class clase { private cadena; clase() { cadena = "Hola"; } public void funcion (HttpServletResponse response) throws IOException, ServletException { 7 PrintWriter object = response.getWriter(); 8 object.println("Hola"); } }</pre>

A partir del código mostrado, entre las líneas numeradas, indique cuál de ellas sería la encargada de imprimir la palabra "Hola" en la pantalla del navegador:

- (a) Línea 5
- (b) Línea 7
- (c) Otra: (Indique la línea)

A partir del código mostrado, entre las líneas numeradas, indique cuál de ellas sería la encargada de imprimir la palabra "Hola" en la pantalla del navegador:

(a) Línea 5

(b) Línea 7

✓ (c) Otra: (Indique la línea)

Solución:	(c) Línea 8
------------------	-------------

[Respuesta_Corta]

Considere los siguientes ficheros de una aplicación JSP con Javabeans:

index.jsp	<pre><html><body><h2>Ejemplo con JavaBeans</h2> <jsp:useBean id="bn" scope="page" class="pk.JB" /> <jsp:setProperty name="EEE" property="test" value="Empty" /> <jsp:getProperty name="FFF" property="test" /> </body></html></pre>
JB.java	<pre>package pk; public class AAA { private String test = "texto"; public JB(BBB) { } public void CCC(String _test) { test = _test; } public String DDD() { return(test); } }</pre>

Indique por qué deben substituirse las cadenas **"AAA"**, **"BBB"**, **"CCC"**, **"DDD"**, **"EEE"**, **"FFF"** para que al acceder a la página **"index.jsp"** se imprima algo en la ventana del navegador ¿Qué se imprime?.

Indique por qué deben substituirse las cadenas "AAA", "BBB", "CCC", "DDD", "EEE", "FFF" para que al acceder a la página "index.jsp" se imprima algo en la ventana del navegador ¿Qué se imprime?.

AAA	JB
BBB	nada
CCC	setTest
DDD	getTest
EEE	bn
FFF	bn

Solución:

Se imprimiría "Empty"

[MultiTest]

Considere una aplicación JSP que contiene, entre otros, los siguientes ficheros:

index.jsp	<pre><html><body><h1>Cabecera de la pagina</h1> <%! String cadena="0"; for(int i=1; i < 10; i++) cadena = cadena +i; %> <form action="servlets/servlet_file" method="post"> Valor: <input type="text" name="value">
 <input type="Submit" name="Enviar"> </form> <jsp:useBean id="miBean" scope="page" class="paquete.jspJB" /> <jsp:getProperty name="miBean" property="variable" /> </body></html></pre>
-----------	--

<code>servlet_ file.java</code>	<pre>import javax.servlet.*; public class servlet_file extends HttpServlet { public void init(ServletConfig conf) throws ServletException { super.init(conf); } public void doPost(HttpServletRequest req, HttpServletResponse res) throws IOException, ServletException { System.out.println("Método Post"); } public void destroy() { }}</pre>
<code>jspJB.java</code>	<pre>package paquete; public class jspJB { public String variable = "cadena"; jspJB() { variable = "init"; } public void setVariable(String _variable) { variable = _variable; } public String getVariable() { return(variable); } }</pre>

Indique las afirmaciones correctas entre las siguientes:

- (a) La etiqueta JSP que alberga el código Java que contiene el bucle `"for"` debería ser `"<%"` en lugar de `"<%!"`.
- (b) No habría sido necesario escribir manualmente el fichero de servlet `"servlet_file.java"`, puesto que el servidor Web (Tomcat) lo habría generado de manera automática.
- (c) La clase `"jspJB"` no sigue la interfaz Javabeen en lo que respecta a la declaración de su variable miembro `"variable"`, pero en general ello no impedirá que funcione correctamente.

Indique las afirmaciones correctas entre las siguientes:

- ✓ (a) La etiqueta JSP que alberga el código Java que contiene el bucle `"for"` debería ser `"<%"` en lugar de `"<%!"`.
- (b) No habría sido necesario escribir manualmente el fichero de servlet `"servlet_file.java"`, puesto que el servidor Web (Tomcat) lo habría generado de manera automática.
- ✓ (c) La clase `"jspJB"` no sigue la interfaz Javabeen en lo que respecta a la declaración de su variable miembro `"variable"`, pero en general ello no impedirá que funcione correctamente.

- [LIMITADA] ¿Qué cláusula SQL se utiliza para hacer consultas a una tabla?:

- ▣ [LIMITADA] ¿Qué cláusula SQL se utiliza para hacer consultas a una tabla?:

SELECT

- [TEST] Una BD tiene 2 tablas: T1 con 2 filas y T2 con 3 filas. ¿Cuántas filas tendrá el resultado de la consulta “SELECT * FROM T1, T2;”?

A) 2

B) 3

C) 6

T1		T2	
col1A	col1B	col2A	col2B
11	1a	21	2a
12	1b	22	2b
		23	2c

- [TEST] Una BD tiene 2 tablas: T1 con 2 filas y T2 con 3 filas. ¿Cuántas filas tendrá el resultado de la consulta “**SELECT * FROM T1, T2;**”?

- A) 2
B) 3
✓ C) 6

T1	
col1A	col1B
11	1a
12	1b

T2	
col2A	col2B
21	2a
22	2b
23	2c

SELECT * FROM T1, T2			
col1A	col1B	col2A	col2B
11	1a	21	2a
11	1a	22	2b
11	1a	23	2c
12	1b	21	2a
12	1b	22	2b
12	1b	23	2c

- [TEST] Dentro del intérprete de psql, la sentencia `dit=> \i file.sql` permite:
 - incluir los datos contenidos en el fichero "file.sql" dentro de la tabla "dit".
 - importar los datos de las tablas y sus estructuras correspondientes contenidos en el fichero "file.sql", dentro de la BBDD seleccionada.
 - leer las sentencias SQL a ejecutar desde el fichero "file.sql".

- [TEST] Dentro del intérprete de psql, la sentencia `dit=> \i file.sql` permite:
 - incluir los datos contenidos en el fichero "file.sql" dentro de la tabla "dit".
 - importar los datos de las tablas y sus estructuras correspondientes contenidos en el fichero "file.sql", dentro de la BBDD seleccionada.
 - ✓ ■ leer las sentencias SQL a ejecutar desde el fichero "file.sql".

- [TEST] ¿Qué método de qué tipo de objeto hay que invocar en Java para ejecutar una sentencia SQL de consulta?
 - Método `executeUpdate()` de `Statement`.
 - Método `executeQuery()` de `Statement`.
 - Método `executeSql()` de `Connection`.

- [TEST] ¿Qué método de qué tipo de objeto hay que invocar en Java para ejecutar una sentencia SQL de consulta?
 - Método executeUpdate() de Statement.
 - ✓ ■ Método executeQuery() de Statement.
 - Método executeSql() de Connection.

- Indique las afirmaciones correcta o incorrectas referidas a SQL o JDBC:
 - (A) Un objeto ResultSet mantiene un cursor que apunta al registro actual. El método ResultSet.next() se usa para moverse al siguiente registro del ResultSet, haciendo el siguiente registro el actual.
 - (B) DROP permite borrar una tabla o una base de datos.
 - (C) ADD INTO usuarios (name,password) VALUES ('user1', 'pass1'); añade un registro a la tabla usuarios.
 - (D) El método executeQuery se usa para operaciones de creación, modificación o borrado de tuplas (registros) de una tabla.

- Indique las afirmaciones correcta o incorrectas referidas a SQL o JDBC:
 - (A) Un objeto ResultSet mantiene un cursor que apunta al registro actual. El método ResultSet.next() se usa para moverse al siguiente registro del ResultSet, haciendo el siguiente registro el actual.
 - (B) DROP permite borrar una tabla o una base de datos.
 - (C) ADD INTO usuarios (name,password) VALUES ('user1', 'pass1'); añade un registro a la tabla usuarios.
 - (D) El método executeQuery se usa para operaciones de creación, modificación o borrado de tuplas (registros) de una tabla.

SOL: A, B

- Si se desea comprobar que se tiene acceso a una BBDD desde JSP, sin ejecutar ninguna sentencia SQL, es necesario utilizar los siguientes objetos de Java:

(A) Socket

(B) DriverManager

(C) Connection

(D) Statement

- Si se desea comprobar que se tiene acceso a una BBDD desde JSP, sin ejecutar ninguna sentencia SQL, es necesario utilizar los siguientes objetos de Java:

(A) Socket

(B) DriverManager

(C) Connection

(D) Statement

SOL: B, C