



XUNTA DE GALICIA
CONSELLERÍA DE FACENDA
Dirección Xeral da Función Pública



***Proceso selectivo para el ingreso en el cuerpo de gestión
de la Xunta de Galicia (subgrupo A2)
escala de gestión de sistemas de información***

Segundo ejercicio

Anexo

NO abra el examen hasta que el tribunal se lo indique

Figura 4. Diagrama PERT

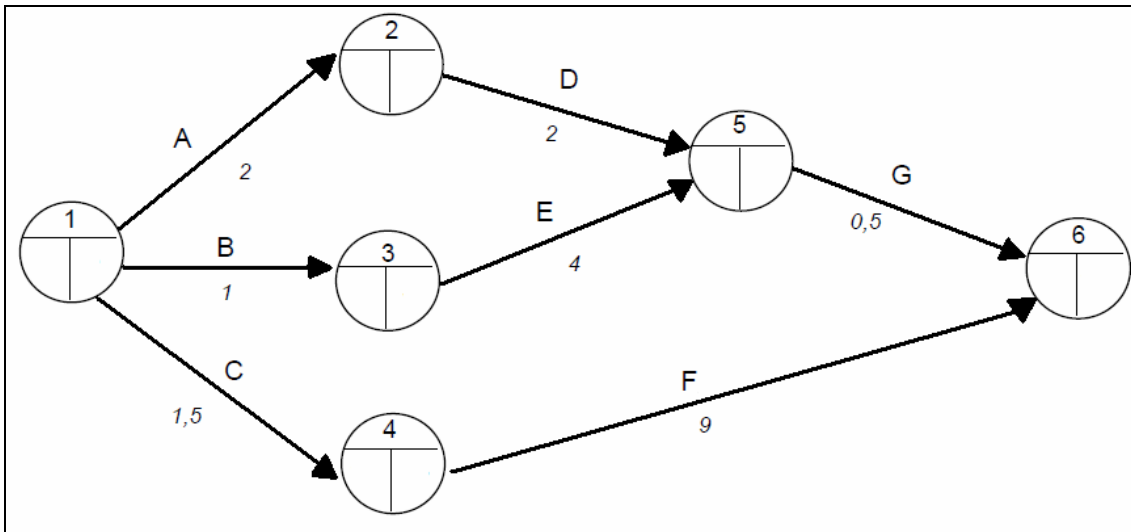


Figura 5 Conexión VPN

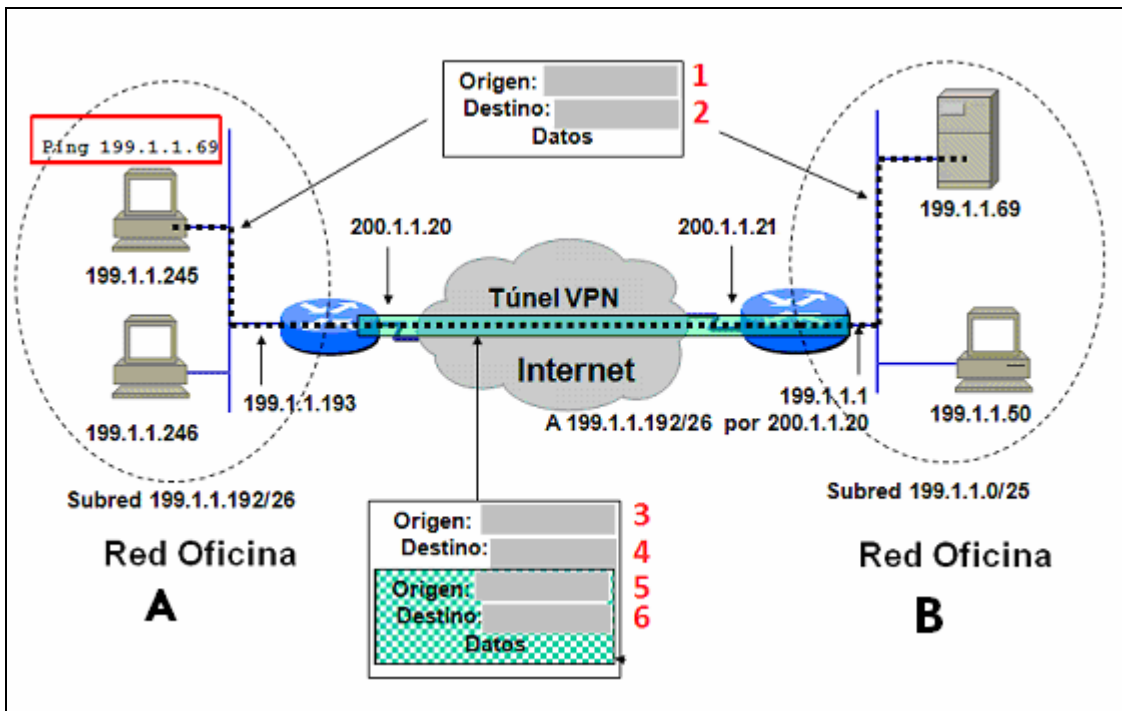


Figura 12.1. El programa pr.c

```

#include <stdio.h>
#include <unistd.h>
main() {
    int i;
    for (i=0;i<3;i++) {
        fork();
        printf("i=%d\n",i); }
}
  
```

Figura 12.2. Salida al ejecutar el programa pr

```
Xunta@comp2379:~/Documentos$ gcc -o pr pr.c
Xunta@comp2379:~/Documentos$ ./pr
i=0
i=0
i=1
i=2
i=1
i=2
i=1
i=2
Xunta@comp2379:~/Documentos$ i=2
i=1
i=2
i=2
i=2
i=2
```

Figura 12.3. Extracto del manual de la función fork()

```
Xunta@comp2379:~/Documentos$ man fork

NAME

    fork -- create a new process

SYNOPSIS

    #include <unistd.h>

    pid_t
    fork(void);

DESCRIPTION

    Fork() causes creation of a new process. The new process (child process)
    is an exact copy of the calling process (parent process) except for the
    following:

        o The child process has a unique process ID.

        o The child process has a different parent process ID (i.e., the
          process ID of the parent process).

        o The child process has its own copy of the parent's descriptors.
          These descriptors reference the same underlying objects, so
          that, for instance, file pointers in file objects are shared
          between the child and the parent, so that an lseek(2) on a
          descriptor in the child process can affect a subsequent read or
          write by the parent. This descriptor copying is also used by
          the shell to establish standard input and output for newly cre-
          ated processes as well as to set up pipes.

        o The child processes resource utilizations are set to 0; see
          setrlimit(2).

RETURN VALUES

    Upon successful completion, fork() returns a value of 0 to the child
    process and returns the process ID of the child process to the parent
    process. Otherwise, a value of -1 is returned to the parent process, no
    child process is created, and the global variable errno is set to indi-
    cate the error.
```

ERRORS

Fork() will fail and no child process will be created if:

- | | |
|----------|---|
| [EAGAIN] | The system-imposed limit on the total number of processes under execution would be exceeded. This limit is configuration-dependent. |
| [EAGAIN] | The system-imposed limit MAXUPRC (<sys/param.h>) on the total number of processes under execution by a single user would be exceeded. |
| [ENOMEM] | There is insufficient swap space for the new process. |

Figura 12.4. Extracto del manual del comando ps

```
Xunta@comp2379:~/Documentos$ man ps
```

NAME

ps - report a snapshot of the current processes.

SYNOPSIS

ps [options]

DESCRIPTION

ps displays information about a selection of the active processes. If you want a repetitive update of the selection and the displayed information, use top(1) instead.

This version of ps accepts several kinds of options:

- 1 UNIX options, which may be grouped and must be preceded by a dash.
- 2 BSD options, which may be grouped and must not be used with a dash.
- 3 GNU long options, which are preceded by two dashes.

Options of different types may be freely mixed, but conflicts can appear. There are some synonymous options, which are functionally identical, due to the many standards and ps implementations that this ps is compatible with.

By default, ps selects all processes with the same effective user ID (euid=EUID) as the current user and associated with the same terminal as the invoker. It displays the process ID (pid=PID), the terminal associated with the process (tname=TTY), the cumulated CPU time in [dd-]hh:mm:ss format (time=TIME), and the executable name (ucmd=CMD). Output is unsorted by default.

The use of BSD-style options will add process state (stat=STAT) to the default display and show the command args (args=COMMAND) instead of the executable name. You can override this with the PS_FORMAT environment variable. The use of BSD-style options will also change the process selection to include processes on other terminals (TTYs) that are owned by you; alternately, this may be described as setting the selection to be the set of all processes filtered to exclude processes owned by other users or not on a terminal. These effects are not considered when options are described as being "identical" below, so -M will be considered identical to Z and so on.

Except as described below, process selection options are additive. The default selection is discarded, and then the selected processes are added to the set of processes to be displayed. A process will thus be shown if it meets any of the given selection criteria.

```

SIMPLE PROCESS SELECTION

-A          Select all processes. Identical to -e.
-N          Select all processes except those that fulfill the specified
           conditions. (negates the selection) Identical to --deselect.
T          Select all processes associated with this terminal. Identical
           to the t option without any argument.
-a          Select all processes except session leaders (see getsid(2))
           and processes not associated with a terminal.
.
.
.

OUTPUT FORMAT CONTROL

These options are used to choose the information displayed by ps. The output
may differ by personality.

-F          extra full format. See the -f option, which -F implies.
-o format is like -o, but preloaded with some default columns. Identical
           to -o pid,format,state,tname,time,command or
           -o pid,format,tname,time,cmd, see -o below.
.
.
.

-f          does full-format listing. This option can be combined with
           many other UNIX-style options to add additional columns. It
           also causes the command arguments to be printed. When used
           with -L, the NLWP (number of threads) and LWP (thread ID)
           columns will be added. See the c option, the format keyword
           args, and the format keyword comm.
j          BSD job control format.
-j         jobs format
l          display BSD long format.
-l         long format. The -y option is often useful with this.
.
.
.

```

Figura 12.5. Extracto de la salida al ejecutar el comando ps con las opciones -Alf.

```

Xunta@comp2379:~/Documentos$ ps -Alf
F S UID          PID  PPID  C  PRI  NI ADDR SZ  WCHAN  STIME TTY          TIME CMD
4 S root           1     0  0  80   0 -   771 select Nov11 ?      00:00:01 /sbin/init
5 S root           2     0  0  75  -5 -    0 kthrea Nov11 ?      00:00:00 [kthreadd]
1 S root           3     2  0 -40  - -    0 migrat Nov11 ?      00:00:00 [migration/0]
1 S root           4     2  0  75  -5 -    0 ksofti Nov11 ?      00:00:01 [ksoftirqd/0]
5 S root           5     2  0 -40  - -    0 watchd Nov11 ?      00:00:00 [watchdog/0]
1 S root           6     2  0 -40  - -    0 migrat Nov11 ?      00:00:00 [migration/1]
1 S root           7     2  0  75  -5 -    0 ksofti Nov11 ?      00:00:00 [ksoftirqd/1]
5 S root           8     2  0 -40  - -    0 watchd Nov11 ?      00:00:00 [watchdog/1]
1 S root           9     2  0  75  -5 -    0 worker Nov11 ?      00:00:00 [events/0]
1 S root          10     2  0  75  -5 -    0 worker Nov11 ?      00:00:00 [events/1]
1 S root          11     2  0  75  -5 -    0 worker Nov11 ?      00:00:00 [khelper]
.
.
.
0 S 1403524762 2009 2003  0  80   0 - 1293 wait  12:41 pts/0    00:00:00 bash
1 S root          2296    2  0  75  -5 -    0 kjourn Nov11 ?      00:00:00 [kjournald]

```

```

1 S root      2297      2 0 75 -5 -      0 kjourn Nov11 ?      00:00:00 [kjournald]
1 S root      2310      2 0 75 -5 -      0 scsi_e 12:47 ?      00:00:00 [scsi_eh_7]
1 S root      2311      2 0 75 -5 -      0 usb_st 12:47 ?      00:00:00 [usb-storage]
0 S 1403524762 2379      1 0 80  0 - 11880 poll  12:48 ?      00:00:01 gedit
5 S daemon    2631      1 0 80  0 - 485 poll  Nov11 ?      00:00:00 /sbin/portmap
5 S root      2653      2 0 75 -5 -      0 worker Nov11 ?      00:00:01 [rpciod/0]
5 S root      2656      2 0 75 -5 -      0 worker Nov11 ?      00:00:00 [rpciod/1]
1 S root      2659      2 0 75 -5 -      0 worker Nov11 ?      00:00:00 [nfsiod]
1 S root      2686      2 0 75 -5 -      0 svc_re Nov11 ?      00:00:00 [lockd]
0 S 1403524762 2798 2009 0 80  0 - 414 n_tty_ 12:55 pts/0      00:00:00 ./pr
0 S 1403524762 2804 2003 0 80  0 - 1293 wait  12:55 pts/1      00:00:00 bash
0 R 1403524762 2856 2804 0 80  0 - 787 -    12:56 pts/1      00:00:00 ps -Alf
.
.
.

```

Figura 12.6. Contenido del fichero pr.s que contiene el código ensamblado del proceso bajo estudio.

```

Xunta@comp2379:~/Documentos$ gcc -S pr
Xunta@comp2379:~/Documentos$ cat pr.s

        .file "pr.c"
        .section      .rodata
.LC0:
        .string "i=%d\n"
        .text
.globl main
        .type main, @function
main:
        leal    4(%esp), %ecx
        andl   $-16, %esp
        pushl  -4(%ecx)
        pushl  %ebp
        movl   %esp, %ebp
        pushl  %ecx
        subl   $36, %esp
        movl   $0, -8(%ebp)
        jmp    .L2
.L3:
        call   fork
        movl   -8(%ebp), %eax
        movl   %eax, 4(%esp)
        movl   $.LC0, (%esp)
        call   printf
        addl   $1, -8(%ebp)
.L2:
        cmpl   $2, -8(%ebp)
        jle    .L3
        addl   $36, %esp
        popl   %ecx
        popl   %ebp
        leal   -4(%ecx), %esp
        ret

```

Figura 18. Diagrama de colaboración

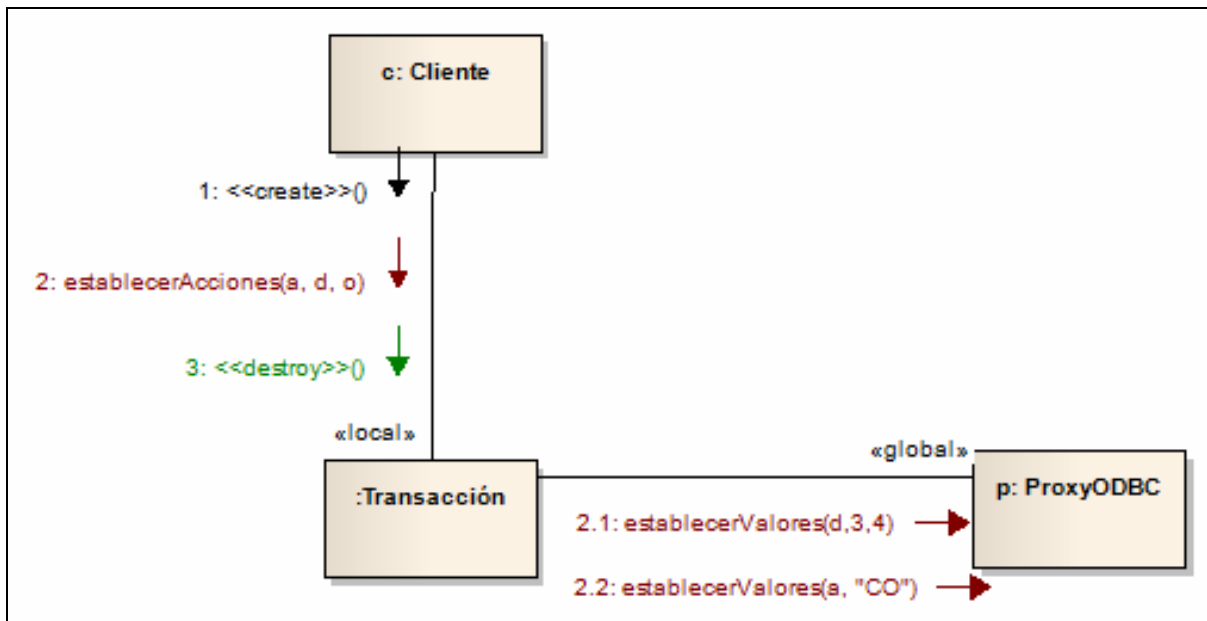


Figura 20. Grafo para averiguar la complejidad ciclométrica

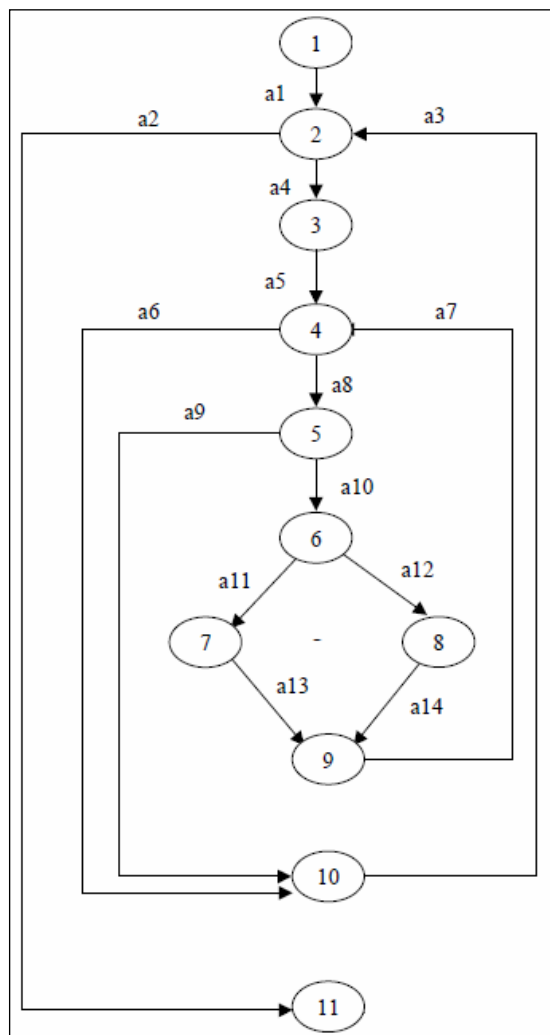


Figura BT3.1. Aspecto de la página de inicio *index.jsp*, visualizada con Firefox 3.0, con el mensaje de error de nombre de usuario no válido

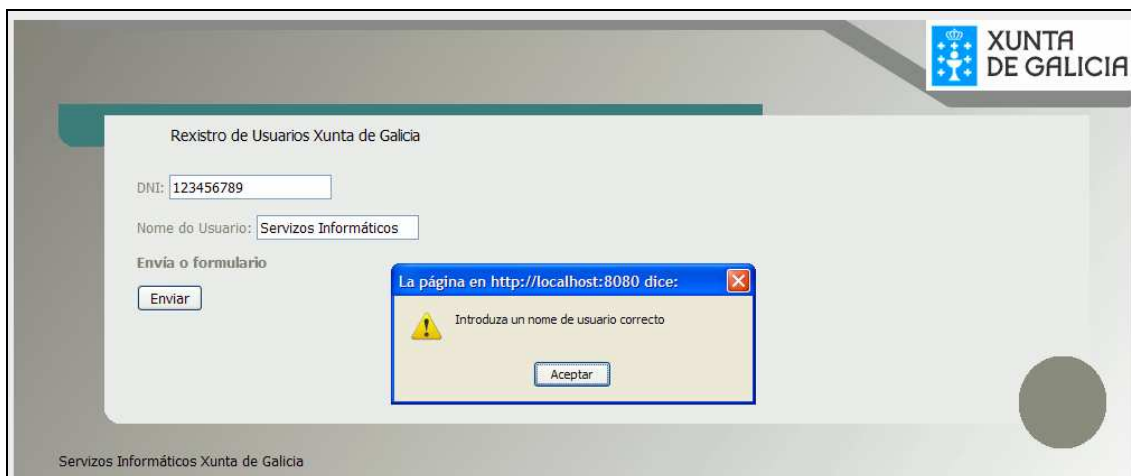


Figura BT3.2. Aspecto de la página devuelta *vista.jsp*, visualizada con Firefox 3.0, cuando el DNI ya está registrado

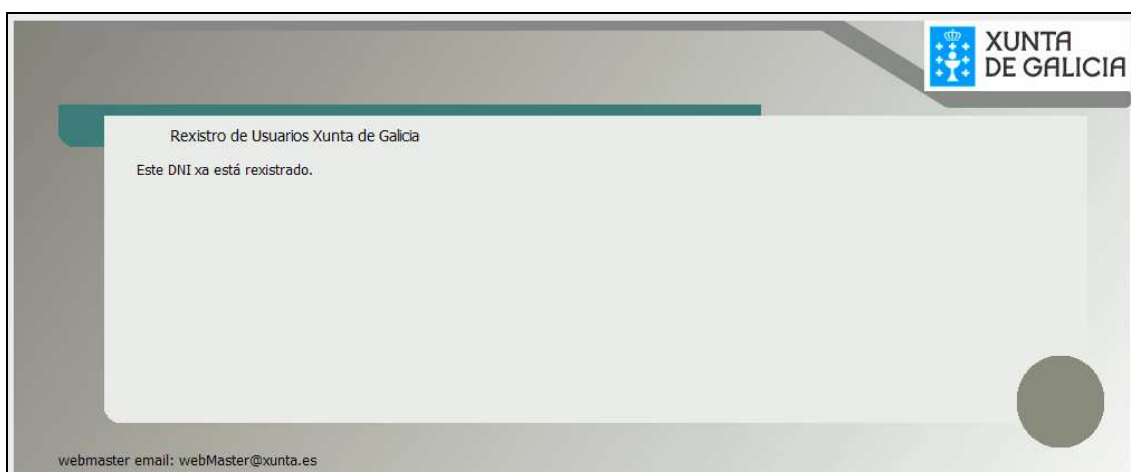


Figura BT3.3. Fragmento del contenido del fichero *estilo.css*

```
Fichero estilo.css → Se presenta solamente una parte del contenido de dicho fichero.  
  
#grande{  
float: left; height: 300px; width: 850px; background: url(imagenes/estilo/grande2.gif) no-repeat bottom  
left; margin-top: 10px; margin-left: 40px; padding-right: 80px; padding-left: 30px;  
}  
  
.titulonegro {  
padding: 2px 20px 0px 30px; margin-top: 7px; color: #000000; font-size: 105%;  
}  
  
.textogrande {  
height: 250px; overflow: auto; padding-right: 10px; padding-top: 0px;  
}  
  
.textogrande p {  
color: #808074; line-height: 1.5em;  
}  
  
.textogrande h2 {  
color: #808074; padding: 0px; margin-top: 5px; margin-bottom: 0px; line-height: 100%;  
}  
  
.pie p {  
font-size: 80%; color: #0000ff; text-align: right;  
}
```


Figura 21. Código del fichero index.jsp en el que faltan contenidos de las líneas 1, 2 y 3

```

Fichero index.jsp
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
    <link rel="stylesheet" type="text/css" href="estilos/estilo.css" id="estilo" />
    <script language="javascript" src="js/validar.js"></script>
  </head>

  <body>
    <div id="contenido">
      <%@include file="jsp/cabecera.jsp"%>
      <div id="grande">
        <p class="titulonegro">Registro de Usuarios Xunta de Galicia</p>
        <div class="textogrande">
          <form name="formUno" [redacted]="registrar" [redacted]="post" onsubmit="[redacted]" />
1         <p>DNI: [redacted] name="dni" id="dniUser" [redacted]</p>
2         <p>Nome do Usuario: [redacted] name="usuario" [redacted]</p>
3         <h2>Envía o formulario</h2>
          <p><input type="submit" value="Enviar"/></p>
        </form>
      </div>
    </div>
    <%@include file="jsp/pie.jsp"%>
  </div>
</body>
</html>

```

Figura 22. Código del fichero validar.js en el que faltan contenidos de las líneas 1, 2, 3 y 4

```

Fichero validar.js

function validar(elem, msg){
  var reExp= /^[g\.\-]{2}Xunta[0-9]{2,5}$/;
1  if([redacted])
    return true;
  else{
2    [redacted]
3    [redacted]
  }
}

function validarAlta(){
4  var nombre=[redacted]('user').value;
  return validar(nombre, "Introduza un nome de usuario correcto");
}

```

Figura 23. Código del fichero web.xml, en el que faltan contenidos de las líneas 1, 2 y 3 y 4, 5, 6 y 7

```

Fichero web.xml

<web-app>
  <context-param>
1    <param-name>masterEmail [redacted]
2    <param-value> [redacted]
3  </context-param>
  <servlet>
4    <servlet-name> [redacted]
5    [redacted]registro.registrarJava</servlet-class>
  </servlet>
  <servlet-mapping>
6    [redacted]Registro de usuario [redacted]
7    [redacted]</url-pattern>
  </servlet-mapping>
</web-app>

```


Figura 25. Código del fichero *vista.jsp*, en el que faltan contenidos de las líneas 1, 2 y 3

```
Fichero vista.jsp
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
    <link rel="stylesheet" type="text/css" href="estilos/estilo.css" id="estilo" />
    <script language="javascript" src="js/validar.js"></script>
  </head>

  <body>
    <div id="contenido">
      <%=include file="cabecera.jsp"%>
      <div id="grande">
        <p class="titulonegro">Registro de Usuarios Xunta de Galicia</p>
1      <div [REDACTED]>
2        <p>[REDACTED]</p>
        </div>
      </div>
3      <p>[REDACTED]</p>
      </div>
    </div>
  </body>
</html>
```