

# 1º de Grado en Ingeniería Informática (Todos los grupos)

## Examen final de Fundamentos de Computadores (Convocatoria de junio)

28 de mayo de 2015

Nombre y apellidos: \_\_\_\_\_ DNI: \_\_\_\_\_

### Instrucciones para realizar el examen (tipo A)

- El tiempo total disponible es de 3 horas (45 minutos para el test).
- No olvidar poner el nombre y apellidos tanto en las hojas de examen como en los folios entregados.
- Utilizar bolígrafo.
- Para las preguntas tipo test, seleccionar una única respuesta en cada cuestión en el lugar habilitado para ello (señalando con una X en la tabla colocada al comienzo del test).
- Cada dos respuestas incorrectas en el test anulan una correcta. Una pregunta sin contestar ni suma ni resta.
- Todos los alumnos deberán entregar tanto las hojas del examen como los folios utilizados al acabar.

### Parte I: tipo test (37.5%; 0.15 puntos por respuesta)

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25
a																									
b																									
c																									

**T1.** El número de **bits** necesarios para almacenar los píxeles de una imagen PPM en formato RGB, con cada valor de R, G y B entre 0 y 255 (sin contar la cabecera) es.

- a) Número de filas \* Número de columnas \* 255
- b) Número de filas \* Número de columnas \* 3 \* 8
- c) Número de filas \* Número de columnas \* 256 \* 3

**T2.** Considérese una CPU que funciona a 2 GHz, ejecutando una secuencia de exactamente 2000 millones de instrucciones, cada una de las cuales se ejecuta en exactamente 3 pasos, cada uno de ellos en un ciclo diferente. Dicha secuencia tardará en ejecutarse exactamente:

- a) 2 segundos.
- b) 3 segundos.
- c) 3 milisegundos.

**T3.** El entero negativo menor que puede representarse en 32 bits es (usando representación en complemento a 2):

- a)  $-2^{32}$ .
- b)  $-2^{31}+1$ .
- c)  $-2^{31}$ .

**T4.** El número 1111 1111 1111 1111 1111 1111 1111 1111 en formato IEEE 754 representa:

- a) NaN (Not a Number).
- b) Menos infinito.
- c) Un número negativo con un valor absoluto muy grande.

**T5.** Elige la única respuesta verdadera de las tres siguientes opciones:

- a) El rango de una representación entera sesgada que utiliza 10 bits es siempre  $[-2^{10}-1; +2^{10}-1]$ , independientemente del sesgo utilizado.

b) El rango de una representación entera sesgada que utiliza 10 bits es siempre  $[-2^9+1; +2^9-1]$ , independientemente del sesgo utilizado.

c) El rango de una representación entera sesgada que utiliza 10 bits siempre tendrá  $2^{10}$  valores distintos, independientemente del sesgo utilizado.

**T6.** Supongamos que necesitamos codificar 63 emoticones para una nueva app de móvil, así como 15 caracteres especiales adicionales. El número mínimo de bits para codificarlos sería de:

- a) 8 bits por cada dígito empleado.
- b) 7 bits por cada dígito empleado.
- c) 6 bits por cada dígito empleado.

**T7.** Para una variable numérica entera de, por ejemplo, 4 bytes, el esquema *little endian* la recupera de tal forma que:

- a) Los valores más significativos están en las direcciones más altas de memoria y los menos significativos en las más bajas.
- b) Los valores más significativos están en las direcciones más bajas de memoria y los menos significativos en las más altas.
- c) El esquema *little endian* sólo se aplica a variables numéricas reales.

**T8.** El número obtenido al desplazar una secuencia de dígitos hexadecimales 2 posiciones a la izquierda (añadiendo ceros por la derecha) es:

- a) El número original multiplicado por  $2^{16}$ .
- b) El número original multiplicado por  $16^2$ .
- c) El número original multiplicado por  $2^{15}$ .

**T9.** Sea el número de 16 bits 81AB, expresado en C2 de 16 bits. Si quisiéramos extenderlo a 32 bits, representando el mismo valor entero, tendríamos el siguiente valor de 32 bits (todas las secuencias de bits expresadas en hexadecimal):

- a) 000081AB
- b) FFFF81AB
- c) 111181AB

**T10.** Para expresar y analizar circuitos lógicos se utiliza una parte de la matemática llamada:

- a) Álgebra de Taylor.
- b) Álgebra de Riemann.
- c) Álgebra de Boole.

**T11.** El mapa de Karnaugh es :

- a) Es la suma de productos (minitérminos) de una función lógica.
- b) Una representación gráfica de una tabla de verdad para una función lógica.
- c) Es el producto de sumas (maxitérminos) de una función lógica.

**T12.** Una salida no determinada en una función lógica significa que:

- a) Valdrá 1 si hay más unos que ceros para las entradas de la función.
- b) Valdrá 1 si hay más ceros que unos para las entradas de la función.
- c) Puede valer 1 o 0 independientemente del valor de las entradas de la función.

**T13.** Sólo uno de los siguientes comandos de Linux se puede considerar un filtro de caracteres. ¿Cuál?

- a) El comando `who`.
- b) El comando `tr`.
- c) El comando `ps`.

**T14.** El comando “`ls p*.a?[!f-z]`”:

- a) Listará todos los archivos cuyo nombre comience con la letra `p` y tienen una extensión que puede ser o bien la `a` o bien una letra entre `f` y `z`.
- b) Listará todos los archivos de la forma `pX.Y`, donde `X` es cualquier secuencia de caracteres (incluida la secuencia vacía) e `Y` es una extensión de tres caracteres que comienza con la letra `a` y acaba con cualquier carácter que no esté entre la `f` y `z`.
- c) En ningún caso listará archivo alguno, puesto que los comodines utilizados codifican una secuencia imposible (el conjunto de cadenas de caracteres que cumplen el patrón es el conjunto vacío).

**T15.** Tras ejecutar la línea de comandos `cat file.txt | grep bye >> out.txt`:

- a) Se escriben las líneas del archivo `file.txt` que contienen la palabra `bye` al final del archivo `out.txt`, si este ya existía, o se produce un error si `out.txt` no existía.
- b) Se escriben las líneas del archivo `file.txt` que contienen la palabra `bye` al final del archivo `out.txt`, creando éste último en caso de que no existiese.
- c) Se escribe la posible salida de error del comando `grep bye`, que toma su entrada de la salida del comando `cat file.txt`, en el archivo `out.txt`.

**T16.** Los principales comandos y programas de un sistema Linux se encuentran en el(los) directorio(s):

- a) `/home`
- b) `/etc` y `/lib`
- c) `/bin` y `/usr/bin`

**T17.** En Linux se denominan filtros:

- a) A una serie de procesos virtuales en *background*.
- b) A todos aquellos comandos que redirigen por defecto la salida estándar a un fichero.
- c) A un repertorio de comandos especialmente diseñado para comunicarse entre sí mediante tuberías.

**T18.** El tercer elemento de un array de enteros de 32 bits que comienza en la etiqueta `enteros`: (e.d., `enteros[2]`, si dicho array empieza en `enteros[0]`) se almacenará exactamente en:

- a) La dirección `X+2`, siendo `X` la dirección correspondiente a la etiqueta `enteros`:
- b) La dirección `X+3`, siendo `X` la dirección correspondiente a la etiqueta `enteros`:
- c) La dirección `X+8`, siendo `X` la dirección correspondiente a la etiqueta `enteros`:

**T19.** En la programación en ensamblador de los procesadores del Intel x86-64 llamamos operandos inmediatos a:

- a) Aquellos operandos que están codificados en la propia instrucción.
- b) Aquellos operandos que al ejecutar la instrucción los encuentra en la dirección de memoria inmediatamente posterior a la que apunta el registro `RDI`.
- c) A los operandos que se encuentran en algún registro de acceso inmediato del procesador.

**T20.** La instrucción de ensamblador `jle .L3` de la arquitectura IA64:

- a) Salta a la instrucción apuntada por la etiqueta `.L3` si el resultado de la última comparación cumplía la condición de que el primer operando era menor o igual que el segundo.
- b) Salta a la subrutina que comienza en la dirección `.L3` (para después volver al encontrar un `ret`).
- c) Compara el resultado de la instrucción anterior con la variable contenida en la dirección `.L3`.

**T21.** Para la configuración mínima de un interfaz de red se debe dotar al sistema operativo de los siguientes valores:

- a) Dirección IP del host, máscara de subred y dirección IP del router de salida.
- b) Dirección MAC (dirección hardware) del interfaz, dirección de puerto del ordenador (habitualmente el

puerto 80).

- c) Dirección IP asignada a la red virtual y dirección del servidor DNS.

**T22.** El protocolo DHCP se usa fundamentalmente para:

- a) Asignar dinámicamente una dirección IP a nuestro interfaz de red.
- b) Conocer dinámicamente la dirección del host principal que se usará como servidor de nombres de dominios.
- c) DHCP significa “Dynamic Host Custom Prompt” y sirve por tanto para apuntar dinámicamente a la cabecera de la red.

**T23.** Señala cual de las siguientes protocolos NO está relacionado con protocolos de red:

- a) IEEE1394.
- b) TCP/IP.
- c) NAT.

**T24.** La dirección 155.54.0.255, en la subred 155.54.0.0/22, se refiere a:

- a) La dirección de red.
- b) La dirección de broadcast.
- c) Una posible dirección del router.

**T25.** Un mismo servidor web en una máquina A sirviendo páginas a dos navegadores distintos que se ejecutan en una misma máquina B:

- a) Realizará sendas comunicaciones desde la IP de A a la misma IP de B, variando el número de puerto en B.
- b) Necesitará que la máquina B tenga al menos dos direcciones distintas, para poder realizar cada una de las dos comunicaciones desde la IP de A a las dos IP distintas de B.
- c) En ningún caso podrá realizar tal tipo de comunicación doble de forma simultánea.

# 1º de Grado en Ingeniería Informática (Todos los grupos)

## Examen final de Fundamentos de Computadores (Convocatoria de junio)

28 de mayo de 2015

Nombre y apellidos: \_\_\_\_\_ DNI: \_\_\_\_\_

### Parte II: cuestiones teórico-prácticas (35%; puntuación indicada en cada apartado)

**C1.** (0.75 puntos) El siguiente ejercicio explora las conversiones de números binarios con y sin signo a números decimales. Vamos a utilizar números de 16 bits ( $n=16$ ). Dado el número  $1010010010010010_2$  (en base 2):

- Si está representado en C2 (complemento a 2), ¿qué número decimal representa?
- Si está representado como un entero sin signo, ¿qué número decimal representa?
- Si está representado en notación sesgada, con sesgo igual a  $2^{16}-1$ , o  $32768_{10}$  en decimal ¿qué número decimal representa?
- Por último, ¿qué número hexadecimal representa?

**C2.** (1.0 punto) Dada la siguiente función:  $F(A, B, C, D) = \sum(0, 2, 5, 7, 8, 10, 13, 15)$

- Dar su expresión mínima utilizando puertas NOR (simplificando por ceros).
- Implementarla utilizando un decodificador y una puerta OR de tamaños adecuados (no hace falta que sean mínimos).
- Implementarla utilizando un multiplexor del tamaño adecuado (no hace falta que sea mínimo).

**C3.** (1.0 punto) Gestión de procesos. Suponiendo un ordenador con una sola CPU con capacidades de multitarea:

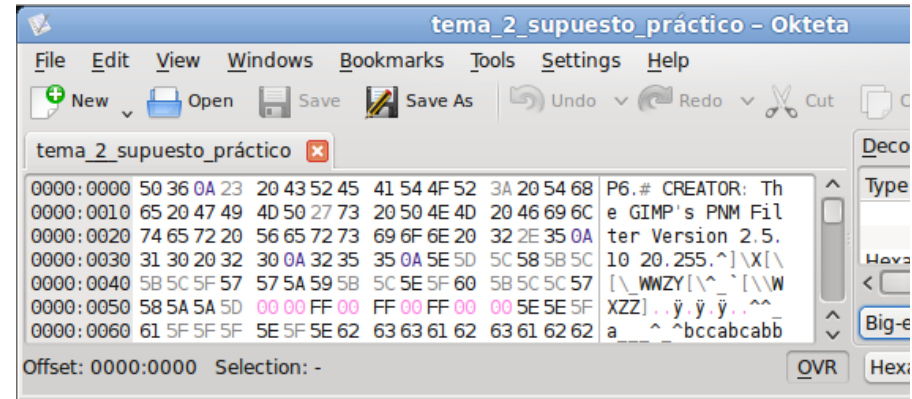
- Explica al menos dos mecanismos mediante los cuales el hardware del ordenador es capaz de interrumpir la tarea que ejecuta actualmente para ejecutar otra bajo el control del S.O.
- ¿Cómo se denomina esa parte del código del S.O. que regula ese cambio de tarea?
- ¿Qué comandos del terminal del S.O. Linux conoces para monitorizar los procesos actualmente activos?

**C4.** (0.75 puntos) Conteste a las siguientes cuestiones sobre redes de computadores:

- Indique las capas en las que se divide la arquitectura de internet, una frase expresando lo más significativo de cada una de ellas en cuanto a su funcionamiento, y según proceda, un ejemplo relativo al hardware, protocolos, y/o programas que correspondan a cada capa.
- URL: ¿Qué significan estas siglas, y de qué partes básicas está compuesta una URL?

### Parte III: ejercicios boletines (27.5%; puntuación indicada en cada apartado)

**P1.** (0.75 puntos) Basándose en los datos que aparecen en la siguiente imagen correspondientes a un fichero abierto con el editor hexadecimal “Okteta”, responde a las siguientes preguntas:



- ¿Qué tipo de fichero se ha abierto? ¿Cómo se sabe?
- ¿Cuál es la longitud total del fichero expresada en bytes?
- ¿Qué están codificando los bytes de los desplazamientos 0000:003A a 0000:003C inclusive?

**P2.** (1.0 punto) Considérese la siguiente sesión con el depurador gdb (los puntos suspensivos [...] indican simplemente que se ha suprimido la parte de la salida que no nos interesa para este ejercicio; aparecen subrayados los comandos tecleados por el usuario; el resto es la salida producida por el terminal). Rellene todos los huecos del texto que va a continuación (indicando en el examen la correspondiente referencia al hueco para cada respuesta):

```
user@host:~/$ gdb ./main
[...]
(gdb) list
[...]
10      int vec[6] = {-10,+20,-30,+40,-50,+60};
11      int main() {
[...]
(gdb) disassemble main
Dump of assembler code for function main:
[...]
0x00000000080483dc <main+56>:  mov     -0x4(%rbp),%eax
0x00000000080483e1 <main+61>:  add     $0x1,%eax
0x00000000080483e4 <main+64>:  mov     %eax, -0x4(%rbp)
0x00000000080483e9 <main+69>:  mov     -0x4(%rbp),%eax
0x00000000080483ee <main+74>:  cmp     $0x5,%eax
0x00000000080483f1 <main+77>:  jle     0x080483c2 <main+30>
0x00000000080483f3 <main+79>:  [...]
[...]
(gdb) x/23b 0x080483dc
0x00000000080483dc <main+56>:  0xXX  0xXX  0xXX  0xXX  0xXX  0xXX  0xXX  0xXX
0x00000000080483e4 <main+64>:  0xXX  0xXX  0xXX  0xXX  0xXX  0xXX  0xXX  0xXX
0x00000000080483ec <main+72>:  0xXX  0xXX  0xXX  0xXX  0xXX  0xXX  0xXX  0xXX
(gdb) x/20b 0x08049660
0x0000000006049660 <vec>:      A      B      C      D      0x14  0x00  0x00  0x00
0x0000000006049668 <vec+8>:    0xe2  0xff  0xff  0xff  0x28  0x00  0x00  0x00
0x0000000006049670 <vec+16>:  0xce  0xff  0xff  0xff  0x3c  0x00  0x00  0x00
```

“El código depurado en la sesión anterior manipula una tabla de \_\_ (a1) \_\_ elementos de tipo entero, que ocupará exactamente \_\_ (a2) \_\_ bytes en memoria, y que comienza en la dirección exacta \_\_ (a3) \_\_. La función principal `main`, por su parte, comienza exactamente en la dirección \_\_ (a4) \_\_. Los valores que aparecen sustituidos en negrita con las letras A, B, C y D eran originalmente, en realidad, los bytes \_\_ (a5) \_\_ (expresar los cuatro bytes en orden, y en formato 0xXX, con XX en hexadecimal). El código desensamblado que aparece en la figura se corresponde, probablemente, con un código de alto nivel que \_\_ (a6) \_\_ (dar aquí una explicación concisa pero completa de lo que hace dicho código). En particular, dentro de ese código la instrucción `cmp` exactamente realiza lo siguiente : \_\_ (a7) \_\_. Un ejemplo de dirección concreta del programa depurado correspondiente con su segmento de datos sería la \_\_ (a8) \_\_, mientras que otra correspondiente a su segmento de texto sería la \_\_ (a9) \_\_. La instrucción ubicada en la dirección 0x80483e1 es de tipo \_\_ (a10) \_\_, mientras que la ubicada en la dirección 0x80483e4 es de tipo \_\_ (a11) \_\_, y la ubicada en 0x080483f1 es de tipo \_\_ (a12) \_\_. Podemos también afirmar que la instrucción `jle`, una vez ubicada en memoria, ocupa exactamente \_\_ (a13) \_\_ bytes. Por último, los bytes que aparecen ocultos en el listado con los valores **0xXX** en realidad se corresponderían con \_\_ (a14) \_\_ (utilizar aquí una concisa frase en castellano que explique el contenido que debería aparecer aquí, en general, sin entrar por supuesto en los valores concretos que tomarían los valores **0xXX**, cuyos valores concretos no nos interesan para contestar a este apartado).”

**Nota:** Los datos enteros de tipo `int` usados en el ejemplo ocupan 4 bytes cada uno, y están almacenados siguiendo el convenio *little-endian*.

**P3.** (1.0 punto) Responda a las siguientes preguntas sobre redes de ordenadores:

- ¿Cuál es la dirección de red y de *broadcast* de un equipo con IP y máscara 192.168.3.1/23?
- Para la red anterior indicar la primera IP válida para su router por defecto.
- Rellene los huecos de cada apartado (c1) y (c2), considerando que ambos comandos Linux se han ejecutado en el mismo equipo:

```
user@host:~$ ifconfig
eth0  Link encap: Ethernet      direcciónHW  00:1A:BC:2E:00:00
      Direc. inet: 155.54.33.1  Difus.:  __ (c1) __      Másc: 255.255.240.0
```

```
user@host:~$ route
Destino    Pasarela      Genmask          Indic  Métric   Ref    Uso   Interfaz
__ (c2) __  *             255.255.240.0    U      0         0     0     eth0
Default    155.54.32.1   0.0.0.0          UG     0         0     0     eth0
```

- Ídem para los apartados (d1), (d2) y (d3): “Si dispongo de una red 155.54.0.0/20 tendré un total (incluyendo posibilidad de router) de hasta (d1) interfaces de red; si configurara entonces todos los equipos de la misma con la máscara 255.255.252.0, obtendría (d2) subredes de (d3) interfaces de red (incluyendo los respectivos routers) cada una.”
- Dada una red con 12 PCs, queremos crear dos subredes privadas de 6 equipos cada una (del rango de direcciones privado 192.168.0.0/16) de tal modo que ambas utilicen el mínimo rango de IPs posibles (elegir las subredes 192.168.0.0 y 192.168.1.0 con la máscara correspondiente). Queremos también que ambas redes tengan salida al exterior a través de un mismo *router*, por lo que cada una de ellas necesitará una IP interna para conectarse al mismo (será asignada la primera posible de cada subred). Se piden, pues, los siguientes datos:

<u>Subred 1:</u> Dirección de red y máscara de red.	<u>Subred 2:</u> Dirección de red y máscara de red.
Dirección de Broadcast.	Dirección de Broadcast.
Router por defecto (IP interna 1).	Router por defecto (IP interna 2).
Direcciones de los equipos.	Direcciones de los equipos.

### Soluciones test:

Tipo A																									
A	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25
a				X			X												X	X	X	X	X		X
b	X	X				X		X	X		X		X	X	X										
c			X		X					X		X				X	X	X						X	

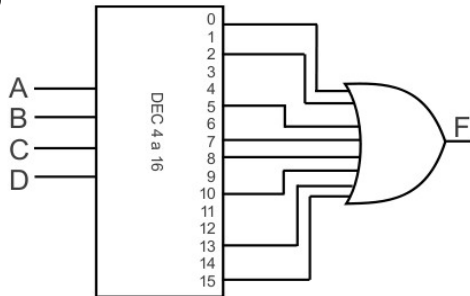
### Solución C1:

- a) -23406 (en base 10).
- b) 42130 (en base 10).
- c)  $42130 - 32768 = 9362$  (en base 10).
- d) A492 (en base 16).

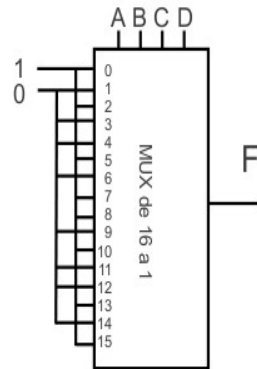
### Solución C2:

$$A) \quad F = (B + \overline{D}) \cdot (\overline{B} + D) = \overline{\overline{(B + \overline{D}) \cdot (\overline{B} + D)}} = \overline{(B + \overline{D}) + (\overline{B} + D)}$$

B)



C)



### Solución C3:

- a) Material de teoría del tema 4, transparencias 24, 29 y 30.
- b) Material de teoría del tema 4, transparencia 23.
- c) Material de teoría del tema 4, transparencias 47 y 48.

### Solución C4:

- a) Material de teoría del tema 6, transparencia 16.
- b) Material de teoría del tema 6, transparencia 36.

### Solución P1:

- a) Es un fichero de mapa de píxeles de tipo PPM. Se sabe porque los dos primeros bytes corresponden a los caracteres ASCII "P6" que son típicos de un fichero PPM.
- b) El campo "gama", que se encuentra tras el tercer separador (valor hexadecimal 0A) nos dice que es una imagen que utiliza un valor máximo de 255 tonos para cada color R, G o B, es decir 3 bytes por cada pixel de la imagen. La imagen está formada por 10 columnas y 20 filas (en ASCII, los valores separados por un espacio en blanco en el segundo campo (tras el segundo separador)). Por lo tanto los bytes de datos del fichero ocuparán  $10 \times 20 \times 3 = 600$  Bytes. La cabecera del fichero acaba en el desplazamiento 0000:0039 inclusive. Ocupa, pues  $39_{16}$  bytes.  $39_{16}$  bytes en hexadecimal son  $3 \times 16 + 9 = 48 + 9 = 57$  bytes (en decimal). La longitud total del fichero es de  $57 + 600 = 657$  Bytes.
- c) Los desplazamientos 0000:003A a 0000:003C están justo a continuación de la cabecera, donde comienza el mapa de píxeles y, en este tipo de fichero .PPM y con esta "gama", cada tres bytes se corresponden a los tres colores fundamentales de un pixel. Así que entre dichos desplazamientos se encuentran los tres canales de color (rojo, verde y azul) del pixel de la primera fila de la imagen, y justo el que está más a la izquierda.

### Solución P2:

- a1) 6.
- a2) 24.
- a3) 0x6049660
- a4) 0x80483dc
- a5) 0xF6 0xFF 0xFF 0xFF
- a6) Incrementa la variable de control de un bucle, comprueba si ésta ha llegado hasta seis y, en caso contrario (es decir, es menor o igual que cinco), realiza un salto hacia atrás para volver a ejecutar otro paso del bucle.
- a7) Comparar el contenido del registro %eax con la constante 5.
- a8) 0x6049670 (por ejemplo).
- a9) 0x80483e4 (por ejemplo).
- a10) Aritmético-lógica.
- a11) Transferencia de datos.
- a12) Salto condicional.
- a13) 2 bytes.
- a14) Se trata del código máquina correspondiente al listado de código desensamblado mostrado anteriormente.

### Solución P3:

- a) Red 192.168.2.0 y broadcast 192.168.3.255.
- b) 192.168.2.1
- c1) 155.54.47.255.
- c2) 155.54.32.0.
- d1) 4094.

d2) 4.

d3) 1022.

e) Subred 1:

*Dirección de red y máscara de red.* 192.168.0.0/28

*Dirección de Broadcast.* 192.168.0.15

*Router por defecto (IP interna 1)* 192.168.0.1

*Direcciones privadas de los equipos* Seis cualquiera del rango 192.168.0.2 a 192.168.0.14

Subred 2:

*Dirección de red y máscara de red.* 192.168.1.0/28

*Dirección de Broadcast.* 192.168.1.15

*Router por defecto (IP interna 2)* 192.168.1.1

*Direcciones privadas de equipos.* Seis cualquiera del rango 192.168.1.2 a 192.168.1.14