UNIVERSIDAD DE ZARAGOZA

Resolución de 4 de diciembre de 2001, de la Universidad de Zaragoza, por la que se convocan pruebas selectivas para el ingreso, por el procedimiento de oposición libre, en plazas vacantes en la plantilla de personal laboral, (Centro de Cálculo, Centro Politécnico Superior, Escuela Universitaria de Ingeniería Técnica Industrial) mediante contratación laboral de carácter fijo.

PROGRAMADOR

SEGUNDO EJERCICIO

Material disponible para la realización de la prueba

Se dispone de un PC con S. O. Windows 98 conectado a la red, con navegadores y emulador de terminales telnet (accesible desde el símbolo del sistema) y de una cuenta en la Worstation UNIX exam2002.unizar.es (Dirección IP 155.210.100.10)

La cuenta es accesible mediante el *username y password* que se han entregado al comienzo de la prueba. El directorio de trabajo (en adelante $HOME) es /export/home/*username*.

La workstation tiene instalado el compilador gcc y su trayectoria, al igual que la del comando make, esta definida en la variable PATH. Se puede utilizar el editor vi.

Ejercicio 1 : Instalación y adecuación de una aplicación bajo S. O. UNIX

Tiempo estimado: 90 Minutos. Puntuación 28 %

En este ejercicio se valorará la documentación de la instalación en la que se expliquen y justifiquen los pasos dados, así como posibles alternativas que se descubran. Si no se consigue terminar alguna de las partes, se valorarán las indicaciones que se den sobre como podría hacerse.

Una empresa ha decidido publicar su catalogo de productos en Internet utilizando un servidor web y su información comercial interna, que servirá para que sus comerciales la consulten desde cualquier lugar con otro servidor web. El acceso a la información comercial deberá ser autentificado. El bajo presupuesto del departamento de informática les obliga a que ambos servidores estén en la misma máquina y accesible bajo nombres distintos, para presentar distinta información a los empleados y a los clientes.

a) Instalar el software Apache versión 1.3.26 en la Workstation. Dicho software se encuentra comprimido (zip) y empaquetado (tar) en el directorio /export/home/SRC.
Descomprimirlo en el directorio $HOME/tmp e instalarlo en el directorio $HOME/apache. Una vez instalado y antes de ejecutarlo, configurarlo para que atienda peticiones por el puerto 80XX donde XX son los dígitos de su username (*) y además, para que el directorio que contiene las páginas web sea $HOME/htdocs . Crear una página de inicio de modo que en ella aparezca su username y un enlace a las páginas de ayuda de la aplicación. No olvide que cada vez que modifique la configuración del servidor, para que los cambios sean efectivos, debe enviarle una señal SIGHUP al proceso padre (o matarlo y reiniciarlo de nuevo)

b) Crear dos estructuras de páginas en dos directorios distintos bajo $HOME/htdocs: interno y externo. Configurar el servidor para crear dos servidores virtuales basados en nombre (misma dirección IP pero nombres distintos, de su elección) de modo que si se accede con un nombre, devuelve la estructura de páginas bajo el direcorio $HOME/htdocs/interno y si se accede con el otro nombre, devuelve estructura bajo el directorio $HOME/htdocs/externo. Crear las páginas de modo que su contenido identifique claramente a que estructura se está accediendo.
Como es un servicio en pruebas, los nombres elegidos para los hosts virtuales no estarán dados de alta en el servicio DNS, por tanto, el PC deberá resolver localmente las direcciones IP para poder acceder e ellos (*). Indique que nombres ha utilizado.

c) Proteger ambas estructuras de directorios de modo que solo se pueda acceder desde máquinas de la subred 155.210.100.0 y autentificando mediante *username* y *password*. Para esto, crear un único fichero de *username/password* en un sitio común para los dos servidores virtuales. Permitir que a una estructura se acceda únicamente con un usuario y a la otra únicamente con otro, distinto del anterior.
¿Como se podría hacer lo anterior para dos grupos distintos en vez de usuarios individuales?.
Si no ha completado la parte b), puede hacer esta parte con el directorio creado en la parte a). Debe dejar constancia de los *usernames, passwords* que utilice para poder comprobar la correcta realización del ejercicio.

Nota:
Los puntos marcados (*) son necesarios para el desarrollo del ejercicio. Si no puede resolverlos, puede consultar al tribunal. Por cada ayuda que se le de, se descontará un 25% del valor total del ejercicio.

Ejercicio 2: Programación en C

Tiempo estimado: 30 Minutos. Puntuación 18 %

Este ejercicio es de programación en C. Siguiendo las especificaciones podrá resolver las cuestiones planteadas aunque no conozca conceptos de comunicaciones.

Codificar en C el tratamiento que un ordenador hace de los paquetes TCP/IP una vez que están listos en el Nivel 3, es decir, formado el paquete, de manera simplificada, con la dirección IP de origen, la de destino y los datos a transmitir, para pasarlos al nivel 2, añadiendo la cabecera correspondiente (de manera simplificada, dirección física de destino y dirección física de origen).

De manera simplificada, una trama de nivel 2 (ethernet en este caso) consiste en:

Dirección Ethernet (MAC) de destino: 6 bytes.
Dirección Ethernet (MAC) de origen: 6 bytes.
Dirección IP de destino: 4 bytes.
Dirección IP de origen: 4 bytes.
Datos: hasta 1500 bytes.

Construir y utilizar las estructuras de datos que se necesiten. Se proponen las siguientes:

```
#define LONGITUD 255 /* Se supone una subred con 254 hosts maximo */

typedef struct
(
unsigned char dir_ether_dest[6];
unsigned char dir_ether_orig[6];
unsigned char dir_ip_dest[4];
unsigned char dir_ip_orig[4];
unsigned char datos[1501]
) trama;

typedef struct
(
unsigned char dir_ip [4];
unsigned char dir_ether[6];
) Arp;

Arp tabla_arp[LONGITUD];  /*----variable global---*/

unsigned char Mi_IP[4];  /* Mi direccion IP */
unsigned char Mi_mascara[4]; /* Mi máscara de subred */
unsigned char Mi_router[4]; /* IP de mi router */
unsigned char Mi_Ether[6];  /* Mi direccion Ethenet */
```

a) Construir una función que averigüe si dos direcciones IP están o no en la misma subred, utilizando la máscara de subred (si coinciden en ambas direcciones IP los bits correspondientes a los unos de la máscara). La máscara de subred consiste en 4 bytes.

```
boolean misma_subred(unsigned char ip_orig[4], unsigned char ip_dest[4],
unsigned char mascara[4])
```

b) Construir una función que dada una dirección IP, consulte una tabla (array) y devuelva la dirección Ethernet (MAC) correspondiente. Dicha tabla está ordenada por dirección IP de modo que la última entrada siempre será la dirección 255.255.255.255, MAC FF:FF:FF:FF:FF:FF. Si se alcanza esa última dirección, se devolverá dicha dirección MAC significando un error: host de destino desconocido (no se implementa el tratamiento de broadcasts). Se supone que dicha tabla (tabla_arp) está creada y contiene las entradas necesarias.

```
void DireccionEthernet(unsigned char ip[4], unsigned ether[6])
```

c) Construir una función que rellene una estructura con una trama ethernet simplificada, con la estructura trama presentada arriba.
Dicha función lo hará de la siguiente manera: Recibe como argumentos los datos y la IP de destino. Con dicha IP deberá determinar la dirección MAC correspondiente: si la IP de destino está en la misma subred, la MAC asociada a dicha IP, si está en otra subred, la MAC asociada a la dirección IP del router.
Devuelve 0 si la MAC obtenida es FF:FF:FF:FF:FF:FF, denotando host inexistente, y 1 en caso contrario. La trama la copia en una estructura qu ese le pasa como argumento.
Se suponen constantes conocidas las direcciones MAC e IP del ordenador, la máscara de subred y la dirección IP del router.
Utilice las funciones construidas en los apartados a) y b).

```
int Construye_trama( unsigned char ip_dest[4], unsigned char
informacion[1501], trama *Mi_trama)
```

Se valorará la construcción de un programa que llame a todas esta funciones y presente por pantalla los campos de la trama construida, o un mensaje de error si no es posible construirla, así como su compilación en la Workstation.

Ejercicio 3: Pregunta sobre Comunicaciones

Tiempo estimado: 20 Minutos. Puntuación 14 %

Se adjuntan unas tramas decodificadas capturadas en una red.
A la vista de las tramas, identificar las máquinas y sus cometidos, los protocolos de nivel 2, 3 y 4. Describir la "conversación" que tiene lugar. Deduzca, con las tramas aportadas el funcionamiento y objetivo del protocolo al que corresponden. ¿Cree que ha funcionado correctamente?

```
- - - - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time
     1 M      [0.0.0.0]         [255.255.255.255]   590 0:00:00.000   0.000.000
Abs. Time              Summary
16/05/2002 11:08:10 a.m. DHCP: Request, Message type: DHCP Discover

DLC:  ----- DLC Header -----
      DLC:
      DLC:  Frame 1 arrived at  11:08:10.0425; frame size is 590 (024E hex) bytes.
      DLC:  Destination = BROADCAST FFFFFFFFFFFF, Broadcast
      DLC:  Source      = Station 00105AAEA08F
      DLC:  Ethertype   = 0800 (IP)
      DLC:
IP: ----- IP Header -----
      IP:
      IP: Version = 4, header length = 20 bytes
      IP: Type of service = 00
      IP:       000. ....   = routine
      IP:       ...0 .... = normal delay
      IP:       .... 0... = normal throughput
      IP:       .... .0.. = normal reliability
      IP:       .... ..0. = ECT bit - transport protocol will ignore the CE bit

      IP:       .... ...0 = CE bit - no congestion
      IP: Total length    = 576 bytes
      IP: Identification  = 1
      IP: Flags           = 0X
      IP:       .0.. .... = may fragment
      IP:       ..0. .... = last fragment
      IP: Fragment offset = 0 bytes
      IP: Time to live    = 20 seconds/hops
      IP: Protocol        = 17 (UDP)
      IP: Header checksum = A4AD (correct)
      IP: Source address      = [0.0.0.0]
      IP: Destination address = [255.255.255.255]
      IP: No options
      IP:
UDP: ----- UDP Header -----
      UDP:
      UDP: Source port      = 68 (Bootpc/DHCP)
      UDP: Destination port = 67 (Bootps/DHCP)
      UDP: Length           = 556
      UDP: Checksum         = 414E (correct)
      UDP: [548 byte(s) of data]
      UDP:
DHCP: ----- DHCP Header -----
      DHCP:
      DHCP: Boot record type        = 1 (Request)
      DHCP: Hardware address type   = 1 (10Mb Ethernet)
      DHCP: Hardware address length = 6 bytes
      DHCP:
      DHCP: Hops                    = 0
      DHCP: Transaction id          = 5CAEA08F
      DHCP: Elapsed boot time       = 5 seconds
      DHCP: Flags                   = 8000
      DHCP:    1... .... .... .... = Broadcast IP datagrams
      DHCP: Client self-assigned IP address   = [0.0.0.0]
      DHCP: Client IP address                 = [0.0.0.0]
      DHCP: Next Server to use in bootstrap   = [0.0.0.0]
      DHCP: Relay Agent                       = [0.0.0.0]
      DHCP: Client hardware address           = 00105AAEA08F
```

```
DHCP:
DHCP: Host name      = ""
DHCP: Boot file name = ""
DHCP:
DHCP: Vendor Information tag = 63825363
DHCP: Message Type                  = 1 (DHCP Discover)
DHCP: Parameter Request List: 14 entries
DHCP:    1 = Client's subnet mask
DHCP:    3 = Routers on the client's subnet
DHCP:   43 = Vendor specific information
DHCP:   54 = Server identifier
DHCP:   60 = Class identifier
DHCP:   67 = Boot File Option
DHCP:   128 = Unknown Option
DHCP:   129 = Unknown Option
DHCP:   130 = Unknown Option
DHCP:   131 = Unknown Option
DHCP:   132 = Unknown Option
DHCP:   133 = Unknown Option
DHCP:   134 = Unknown Option
DHCP:   135 = Unknown Option
DHCP: Maximum message size          = 1260
DHCP: Unidentified tag 93
DHCP: Unidentified tag 94
DHCP: Class identifier              =
505845436C69656E743A417263683A30303030303A554E44493A30303230303
DHCP:
```

```
- - - - - - - - - - - - - - - - - - - - - - Frame 2 - - - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time
     2        [155.210.208.17]  [255.255.255.255]   342 0:00:00.002   0.002.151
Abs. Time              Summary
16/05/2002 11:08:10 a.m. DHCP: Reply, Message type: DHCP Offer

DLC:  ----- DLC Header -----
      DLC:
      DLC:  Frame 2 arrived at  11:08:10.0447; frame size is 342 (0156 hex) bytes.
      DLC:  Destination = BROADCAST FFFFFFFFFFFF, Broadcast
      DLC:  Source      = Station 0002A5ADB197
      DLC:  Ethertype   = 0800 (IP)
      DLC:
IP: ----- IP Header -----
      IP:
      IP: Version = 4, header length = 20 bytes
      IP: Type of service = 00
      IP:       000. ....   = routine
      IP:       ...0 .... = normal delay
      IP:       .... 0... = normal throughput
      IP:       .... .0.. = normal reliability
      IP:       .... ..0. = ECT bit - transport protocol will ignore the CE bit
      IP:       .... ...0 = CE bit - no congestion
      IP: Total length    = 328 bytes
      IP: Identification  = 36171
      IP: Flags           = 0X
      IP:       .0.. .... = may fragment
      IP:       ..0. .... = last fragment
      IP: Fragment offset = 0 bytes
      IP: Time to live    = 128 seconds/hops
      IP: Protocol        = 17 (UDP)
      IP: Header checksum = 4076 (correct)
      IP: Source address      = [155.210.208.17]
      IP: Destination address = [255.255.255.255]
      IP: No options
      IP:
UDP: ----- UDP Header -----
      UDP:
      UDP: Source port      = 67 (Bootps/DHCP)
      UDP: Destination port = 68 (Bootpc/DHCP)
      UDP: Length           = 308
      UDP: Checksum         = B247 (correct)
      UDP: [300 byte(s) of data]
      UDP:
DHCP: ----- DHCP Header -----
      DHCP:
      DHCP: Boot record type        = 2 (Reply)
      DHCP: Hardware address type   = 1 (10Mb Ethernet)
      DHCP: Hardware address length = 6 bytes
      DHCP:
      DHCP: Hops                    = 0
      DHCP: Transaction id          = 5CAEA08F
      DHCP: Elapsed boot time       = 0 seconds
      DHCP: Flags                   = 0000
      DHCP:    0... .... .... .... = No broadcast
      DHCP: Client self-assigned IP address   = [0.0.0.0]
      DHCP: Client IP address                 = [155.210.208.71]
      DHCP: Next Server to use in bootstrap   = [155.210.208.17]
      DHCP: Relay Agent                       = [0.0.0.0]
      DHCP: Client hardware address           = 00105AAEA08F
      DHCP:
      DHCP: Host name      = ""
      DHCP: Boot file name = ""
      DHCP:
      DHCP: Vendor Information tag = 63825363
      DHCP: Message Type                  = 2 (DHCP Offer)
      DHCP: Subnet mask = [255.255.255.0]
      DHCP: Address Renewel interval      = 21600 (seconds)
      DHCP: Address Rebinding interval    = 37800 (seconds)
      DHCP: Request IP address lease time = 43200 (seconds)
      DHCP: Server IP address           = [155.210.208.17]
      DHCP: Gateway address           = [155.210.208.254]
      DHCP: Class identifier              = 505845436C69656E7400
      DHCP:
```

```
- - - - - - - - - - - - - - - - - - - - - - Frame 3 - - - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time
     3        [0.0.0.0]         [255.255.255.255]   590 0:00:03.954   3.952.219
Abs. Time              Summary
16/05/2002 11:08:13 a.m. DHCP: Request, Message type: DHCP Request


DLC:  ----- DLC Header -----
      DLC:
      DLC:  Frame 3 arrived at  11:08:13.9969; frame size is 590 (024E hex) bytes.
      DLC:  Destination = BROADCAST FFFFFFFFFFFF, Broadcast
      DLC:  Source      = Station 00105AAEA08F
      DLC:  Ethertype   = 0800 (IP)
      DLC:
IP: ----- IP Header -----
      IP:
      IP: Version = 4, header length = 20 bytes
      IP: Type of service = 00
      IP:       000. ....   = routine
      IP:       ...0 .... = normal delay
      IP:       .... 0... = normal throughput
      IP:       .... .0.. = normal reliability
      IP:       .... ..0. = ECT bit - transport protocol will ignore the CE bit
      IP:       .... ...0 = CE bit - no congestion
      IP: Total length    = 576 bytes
      IP: Identification  = 2
      IP: Flags           = 0X
      IP:       .0.. .... = may fragment
      IP:       ..0. .... = last fragment
      IP: Fragment offset = 0 bytes
      IP: Time to live    = 20 seconds/hops
      IP: Protocol        = 17 (UDP)
      IP: Header checksum = A4AC (correct)
      IP: Source address      = [0.0.0.0]
      IP: Destination address = [255.255.255.255]
      IP: No options
      IP:
UDP: ----- UDP Header -----
      UDP:
      UDP: Source port      = 68 (Bootpc/DHCP)
      UDP: Destination port = 67 (Bootps/DHCP)
      UDP: Length           = 556
      UDP: Checksum         = 380E (correct)
      UDP: [548 byte(s) of data]
      UDP:
DHCP: ----- DHCP Header -----
      DHCP:
      DHCP: Boot record type        = 1 (Request)
      DHCP: Hardware address type   = 1 (10Mb Ethernet)
      DHCP: Hardware address length = 6 bytes
      DHCP:
      DHCP: Hops                    = 0
      DHCP: Transaction id          = 5CAEA08F
      DHCP: Elapsed boot time       = 5 seconds
      DHCP: Flags                   = 8000
      DHCP:    1... .... .... .... = Broadcast IP datagrams
      DHCP: Client self-assigned IP address   = [0.0.0.0]
      DHCP: Client IP address                 = [0.0.0.0]
      DHCP: Next Server to use in bootstrap   = [0.0.0.0]
      DHCP: Relay Agent                       = [0.0.0.0]
      DHCP: Client hardware address           = 00105AAEA08F
      DHCP:
      DHCP: Host name      = ""
      DHCP: Boot file name = ""
      DHCP:
      DHCP: Vendor Information tag = 63825363
      DHCP: Message Type                  = 3 (DHCP Request)
      DHCP: Request specific IP address   = [155.210.208.71]
      DHCP: Parameter Request List: 14 entries
      DHCP:    1 = Client's subnet mask
      DHCP:    3 = Routers on the client's subnet
      DHCP:   43 = Vendor specific information
      DHCP:   54 = Server identifier
      DHCP:   60 = Class identifier
      DHCP:   67 = Boot File Option
      DHCP:   128 = Unknown Option
```

```
      DHCP:    129 = Unknown Option
      DHCP:    130 = Unknown Option
      DHCP:    131 = Unknown Option
      DHCP:    132 = Unknown Option
      DHCP:    133 = Unknown Option
      DHCP:    134 = Unknown Option
      DHCP:    135 = Unknown Option
      DHCP: Maximum message size          = 1260
      DHCP: Server IP address           = [155.210.208.17]
      DHCP: Unidentified tag 93
      DHCP: Unidentified tag 94
      DHCP: Class identifier              =
505845436C69656E743A417263683A30303030303A554E44493A303032303031
      DHCP:
```

```
- - - - - - - - - - - - - - - - - - - - - - Frame 4 - - - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time
     4        [155.210.208.17]  [255.255.255.255]   342 0:00:03.957   0.002.736
Abs. Time              Summary
16/05/2002 11:08:13 a.m. DHCP: Reply, Message type: DHCP Ack

DLC:  ----- DLC Header -----
      DLC:
      DLC:  Frame 4 arrived at  11:08:13.9996; frame size is 342 (0156 hex) bytes.
      DLC:  Destination = BROADCAST FFFFFFFFFFFF, Broadcast
      DLC:  Source      = Station 0002A5ADB197
      DLC:  Ethertype   = 0800 (IP)
      DLC:
IP: ----- IP Header -----
      IP:
      IP: Version = 4, header length = 20 bytes
      IP: Type of service = 00
      IP:       000. ....   = routine
      IP:       ...0 .... = normal delay
      IP:       .... 0... = normal throughput
      IP:       .... .0.. = normal reliability
      IP:       .... ..0. = ECT bit - transport protocol will ignore the CE bit
      IP:       .... ...0 = CE bit - no congestion
      IP: Total length    = 328 bytes
      IP: Identification  = 36179
      IP: Flags           = 0X
      IP:       .0.. .... = may fragment
      IP:       ..0. .... = last fragment
      IP: Fragment offset = 0 bytes
      IP: Time to live    = 128 seconds/hops
      IP: Protocol        = 17 (UDP)
      IP: Header checksum = 406E (correct)
      IP: Source address      = [155.210.208.17]

      IP: Destination address = [255.255.255.255]
      IP: No options
      IP:
UDP: ----- UDP Header -----
      UDP:
      UDP: Source port      = 67 (Bootps/DHCP)
      UDP: Destination port = 68 (Bootpc/DHCP)
      UDP: Length           = 308
      UDP: Checksum         = 1B2C (correct)
      UDP: [300 byte(s) of data]
      UDP:
DHCP: ----- DHCP Header -----
      DHCP:
      DHCP: Boot record type        = 2 (Reply)
      DHCP: Hardware address type   = 1 (10Mb Ethernet)
      DHCP: Hardware address length = 6 bytes
      DHCP:
      DHCP: Hops                    = 0
      DHCP: Transaction id          = 5CAEA08F
      DHCP: Elapsed boot time       = 0 seconds
      DHCP: Flags                   = 0000
      DHCP:    0... ....  .... .... = No broadcast
      DHCP: Client self-assigned IP address   = [0.0.0.0]
      DHCP: Client IP address                 = [155.210.208.71]
      DHCP: Next Server to use in bootstrap   = [0.0.0.0]
      DHCP: Relay Agent                       = [0.0.0.0]
      DHCP: Client hardware address           = 00105AAEA08F
      DHCP:
      DHCP: Host name      = ""
      DHCP: Boot file name = ""
      DHCP:
      DHCP: Vendor Information tag = 63825363
      DHCP: Message Type                    = 5 (DHCP Ack)
      DHCP: Address Renewel interval        = 21600 (seconds)
      DHCP: Address Rebinding interval      = 37800 (seconds)
      DHCP: Request IP address lease time = 43200 (seconds)
      DHCP: Server IP address           = [155.210.208.17]
      DHCP: Subnet mask = [255.255.255.0]
      DHCP: Gateway address             = [155.210.208.254]
      DHCP: Class identifier              = 505845436C69656E7400
  DHCP:
```

Ejercicio 4: Modelo de datos de un sistema

Tiempo estimado: 30 Minutos. Puntuación 18 %

Una empresa contrata empleados para determinados puestos de trabajo. A efectos de este supuesto dichos puestos de trabajo (no los empleados que los ocupan), permanecen estables en cuanto a número, tipo y demás atributos.

Cada empleado pertenece a una categoría profesional (peón, oficial, encargado, supervisor, administrativo, etc.). Algunas categorías tienen una jornada de trabajo de 8 horas y sólo pueden ser desempeñados bajo esa jornada, mientras que otros tienen la posibilidad, a elección del empleado y con el visto bueno de la empresa, de ser desempeñados también en jornadas especiales (media jornada, jornada de mañana, etc.). Determinada antigüedad en la categoría otorga, con carácter general, el carácter de senior aunque a algunos empleados este carácter puede ser concedido antes de cumplir esa antigüedad.

La nómina se calcula sumando retribuciones dependientes de la categoría, la jornada, el carácter senior más un complemento en función del puesto de trabajo ocupado.

Se pretende crear un sistema de información que a través de los procedimientos administrativos oportunos (cuya definición, representación o diseño no son objeto de este supuesto) permita conocer la estructura de personal (puestos de trabajo ocupados, número de empleados por categoría, jornada, carácter senior, etc.) en cualquier momento actual o pasado así como calcular las retribuciones a pagar (lo que se hace en función del estado de cada empleado el último día del mes) en el momento actual o en cualquier mes anterior.

Reglas de gestión:

1.- La condición de contratado (es decir el alta y la baja en el contrato), la categoría, la jornada y el carácter "senior" deben de contar con un histórico que refleje las situaciones previas.
2.- El complemento salarial atribuido al puesto de trabajo debe de contar también con un histórico.
3.- Un empleado no puede ocupar simultáneamente (aunque sí en periodos sucesivos) dos puestos de trabajo distintos. Si se produce un alta de un empleado que ya ocupaba otro puesto, se entiende que causa baja en el puesto anterior el dia de antes.
4.- Los cambios de categoría ocasionan la pérdida de la condición "senior" que podrá volverse a obtener tras el plazo requerido.

Se pide:

a) Identificar los atributos necesarios para mantener el sistema de información.
b) Representar el modelo de datos en tercera forma normal
c) Dibujar el diagrama Entidad-Relación del modelo de datos.
d) En el caso en que se haya introducido alguna redundancia de datos:
   d.1 Describir la redundancia
   d.2 Proponer un método para eliminarla
   d.3 Describir un algoritmo para que en el procedimiento de gestión de datos dicha redundancia no ocasione incoherencias en los datos.

Ejercicio 5: Cuestiones SQL

Tiempo estimado: 30 Minutos. Puntuación 14 %

En las cuestiones que continúan, un "script" indica un conjunto de uno o varios comandos SQL que hacen el cálculo solicitado. El usuario tiene permiso para CREATE, DROP, SELECT, UPDATE, INSERT y DELETE.
Las reglas de gestión aplicables a los siguientes supuestos son las enunciadas en el ejercicio 4.

a) En base a la siguiente tabla de "contrataciones"

```
Tabla CONTRATACIONES:

IdEmpleado
IdPuesto
IndicadorContratacion  // A=alta en el contrato, B=baja en el contrato
FechaComienzoSituacion // Fecha en que comienza la situación de
                       //contratación
```

Escribir un script para calcular el número de empleados están contratados en determinada fecha ubicada en la tabla:

```
Tabla FECHA_REFERENCIA:

FechaCalculo
```

b) Escribir un script para calcular las categorías que en determinada fecha tienen más de 10 empleados

```
Tabla EMPLEADOS_CATEGORIAS

IdEmpleado
IdCategoria
IndicadorContratacion  // A=alta en el contrato, B=baja en el contrato
FechaComienzoSituacion // Fecha en que comienza la situación de
                       //contratación
```

La fecha de cálculo está ubicada en la misma tabla FECHA_REFERENCIA

c) Escribir un script para determinar el tiempo máximo en días contiguos en que ha estado contratado cualquier empleado

Tabla sobre la que hacer el cálculo:

```
Tabla EMPLEADO_DIA

IdEmpleado
IdDia  //es un número correlativo para cada día que comienza con el
       //numero 1 que corresponde al 01.01.2000
```

Ejercicio 6: Cuestiones Shell UNIX

Tiempo estimado: 15 Minutos. Puntuación 8 %

Algunos de los siguientes comandos de la **shell ksh** pueden ser de utilidad en este ejercicio.

cat, cut, date, df, find, grep, ls, more, paste, pr, pwd, read, sed, sort, while, test, wc

Se valorará que el resultado haya sido almacenado y verificado en la *workstation.*

a) Cuestiones de manipulación de ficheros

a.1 Buscar todos los ficheros tipo *.txt que estén en un directorio o en subdirectorios del mismo y colocar el resultado en un fichero.

a.2 A partir del anterior. construir un fichero que tenga el tamaño y el nombre completo (path completo) de los ficheros localizados en el apartado anterior

a.3 Ordenar el fichero del apartado anterior de mayor a menor tamaño.

b) Se tiene una lista de nombres de ficheros, almacenados uno por línea en un fichero. Construir un script que muestre los nombres de los ficheros que contienen en su interior determinada cadena de caracteres.

c) Manipulación de ficheros

c.1 Se tienen dos ficheros: numero.dat y palabras.dat. Cada uno de ellos con diez registros, contiendo una número y una palabra respectivamente. Construir un nuevo fichero con diez registros formados cada uno de ellos por el número y la palabra que ocupan el mismo lugar, separados ambos campos por un tabulador.

c.2 Buscar en un fichero las líneas que contengan dos palabras (en cualquier orden) que se den. No importa que la "caja" (mayúsculas o minúsculas) del fichero coincidan con las de la plantilla de búsqueda.