

**NO ABRA ESTE CUADERNILLO
HASTA QUE SE LE INDIQUE**



UNIVERSIDAD DE GRANADA

**PRUEBAS SELECTIVAS PARA INGRESO EN LA
ESCALA BÁSICA DE INFORMÁTICA**

(Resolución de 20 de julio de 2020, BOE nº 208 de 01-08-2020)

SEGUNDA PRUEBA

SUPUESTO A

La red informática de un Campus Universitario tiene la topología del siguiente diagrama:

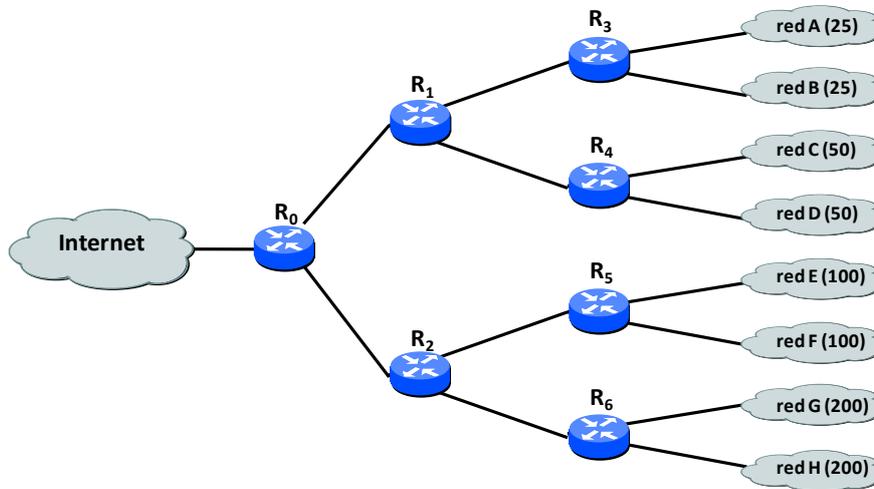


Figura 1: Arquitectura de Red

En la imagen se observan 7 routers (R_0, R_1, \dots, R_6), y ocho LAN's correspondientes cada una de ellas con 8 edificios del Campus. Entre paréntesis se indican los números de nodos de red que cada edificio tiene en operación en su LAN. Toda la red está conectada hacia internet a través de R_0 . Considere que toda la infraestructura de telecomunicación implicada en el diagrama de la figura 1 existe físicamente y está implementada a nivel hardware con la estructura representada.

El rango de direcciones TCP/IP que se dispone para toda la red del Campus es $168.168.168.0/22$. Está distribuido a nivel lógico como se representa en la siguiente figura.

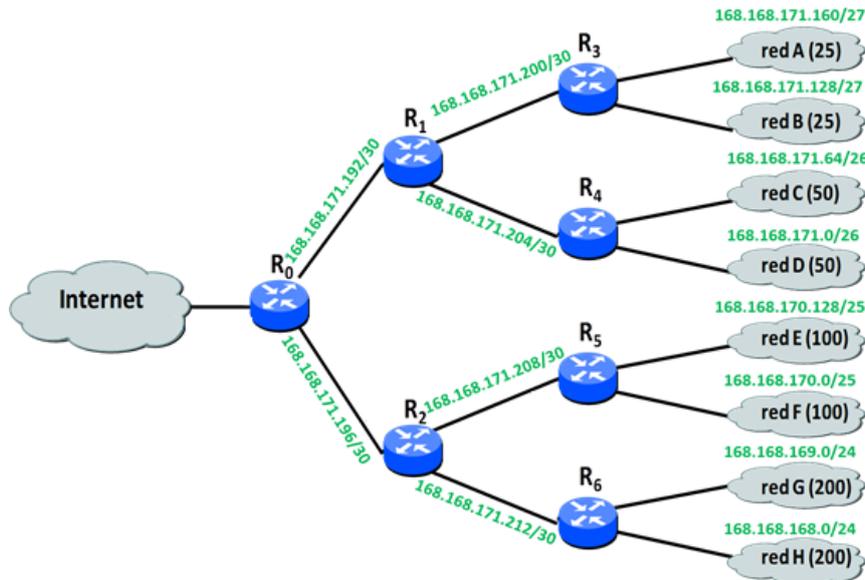


Figura 2: Asignación de direccionamiento IP

Cada router de la red tiene tres interfaces denominados como se representa en la figura 3. Según la misma, el interfaz de la izquierda siempre es el Eth0, el de la derecha-arriba siempre es el Eth1 y el de la derecha-abajo el Eth2.

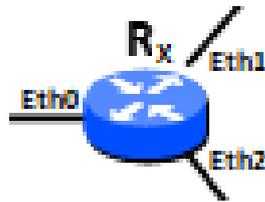


Figura 3: Denominación de los interfaces de los routers de la Red

Se le pide:

1.- Se pretende poner en marcha la red completa a nivel lógico en base a la información técnica que se le ha facilitado para lo cual deberá comenzar con una asignación y configuración de direccionamiento a todos los routers de la figura 1. En primer lugar, explique justificadamente qué procedimiento técnico seguirá para realizar dicha configuración de direccionamiento IP en los routers. Seguidamente, relacione en una tabla con el siguiente formato, el direccionamiento IP que utilizará para la configuración de los interfaces de los routers de la red. Se le fija el criterio de que, en cada subred, a los routers siempre se les asignan la primera dirección IP libre posible según el criterio diseñado en la figura 2.

Router	Interfaz	IP	Máscara
--------	----------	----	---------

2.- Imagínese la situación en la que se ha detectado un problema de conectividad entre un nodo del Edificio de la Red H y uno del Edificio de la Red A por un problema ya identificado de routing. Suponiendo que las tablas de encaminamiento correspondientes a R₁ y R₂ fuesen las siguientes, explique justificadamente cuál es el error que hay (En R₃ y R₆ se ha verificado que no hay ningún problema).

ROUTER R₁

Destino	Máscara	Sig. Salto (interfaz)	Comentario
168.168.171.0	/25	R ₄ (168.168.171.206)	Hacia redes C y D
168.168.171.128	/25	R ₃ (168.168.171.205)	Hacia redes A y B
168.168.171.200	/30	*	Conexión directa
168.168.171.204	/30	*	Conexión directa
default	-	R ₀ (168.168.171.193)	Hacia Internet y otras subredes

ROUTER R₂

Destino	Máscara	Sig. Salto (interfaz)	Comentario
168.168.168.0	/23	R ₆ (168.168.171.214)	Hacia redes G y H
168.168.170.0	/24	R ₅ (168.168.171.210)	Hacia redes E y F
168.168.171.208	/30	*	Conexión directa
168.168.171.212	/30	*	Conexión directa
default	-	R ₀ (168.168.171.197)	Hacia Internet y otras subredes

3.- Considere la situación en la que el enlace de datos que une el interfaz Eth1 de R₁ con el Eth0 de R₃ ubicados en sus correspondientes CPDs (Centro de Proceso de Datos o Data Centers), está construido en base a fibra óptica MM/OM3 y los demás elementos de red necesarios siguiendo el paradigma de los sistemas de cableado estructurado y con la estructura de red descrita de la figura 4. Suponga que le indican que los usuarios de la Red A y la Red B no pueden conectarse a Internet, aunque sí pueden comunicarse entre ellos. Los usuarios de las redes C y D sí tienen conectividad a Internet. Se descartan problemas de configuración lógica en los equipos finales de la red y en los routers, puesto que se ha verificado expresamente. Para identificar la naturaleza del problema descrito, ¿qué operaciones de diagnóstico de red realizaría y en qué herramientas se apoyaría?

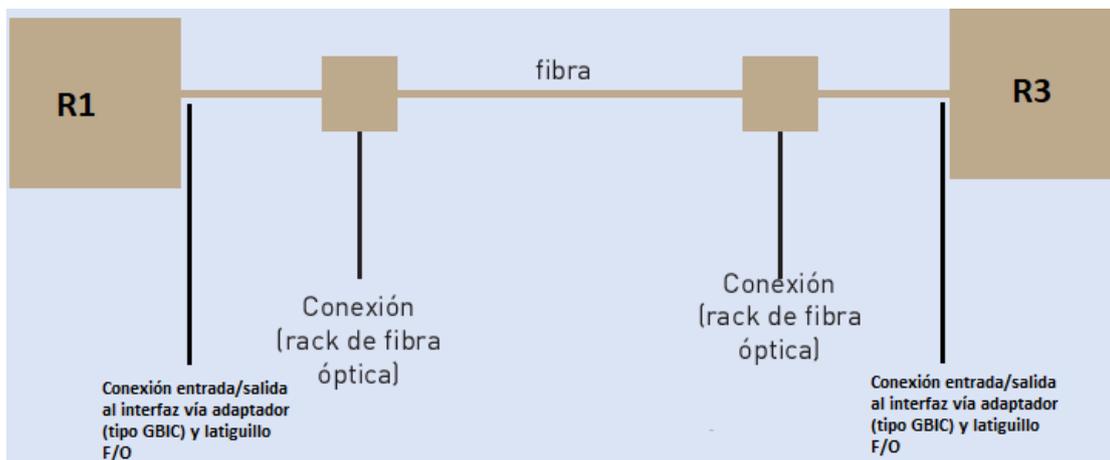


Figura 4: Estructura de conexión entre los routers R1 y R3

FIN DEL SUPUESTO A

SUPUESTO B

En el Instituto de Psicología Aplicada y Mindfulness (conocido por sus siglas IPAMI, con un dominio en Internet *ipami.es*) se instala un sistema informático que permite a sus trabajadores tener una cuenta de usuario con la que pueden (entre otras utilidades como la del uso de correo electrónico, etc.) abrir sesión en equipos con Microsoft Windows. Estos usuarios se dividen en grupos, según el departamento al que pertenecen, los cuales tienen acceso a una unidad de red en la que conjuntamente trabajan y comparten ficheros todos los miembros de un grupo, y sólo los miembros de ese grupo.

Tenemos en funcionamiento el software Samba, en un servidor Linux (Debian versión 9.4), al que vamos a llamar *autentica* (nombre de red *autentica.ipami.es*) que nos provee un servicio de autenticación, donde definimos los usuarios de nuestro sistema y los grupos a los que pertenecen estos usuarios.

En un servicio de almacenamiento, localizado en una cabina en red (*cenamiento.ipami.es*), se definen los volúmenes que proporcionan la unidad de red a cada grupo de usuarios. Este servicio puede ser gestionado y consultado usando una conexión SSH desde el servidor *autentica*, que accediendo con la cuenta “alma” del servidor *cenamiento*, nos permite escribir comandos como:

- `ssh alma@cenamiento.ipami.es "volume show -volume z_$(grupo) -fields size,used,percent-used"; #` donde `$(grupo)` es una variable SHELL que contiene el nombre de uno de los grupos y `z_$(grupo)` es el nombre asignado al volumen correspondiente a cada grupo de trabajo. Este comando nos da como respuesta una salida de varias líneas, entre las cuales, una de ellas tiene la forma “`cenamiento_svm z_contabilidad 250GB 224.5GB 94%`”. En este ejemplo el grupo `contabilidad` tiene su unidad de red al 94% de capacidad.

Disponemos de una API, que nos proporciona una serie de comandos. Suponiendo que `$(grupo)` es una variable que contiene el nombre de uno de los grupos de usuarios, los comandos tienen esta sintaxis:

- `auth-tool group listmembers $(grupo); #` lista los miembros de un grupo, uno por línea.
- `auth-tool user list; #` nos proporciona la lista total de usuarios en el sistema.
- `auth-tool group show $(grupo); #` proporciona toda la información del grupo en varias líneas, entre ellas una que comienza por “`description:`” y continúa con una descripción del grupo de usuarios `$(grupo)`.
- `auth-tool group list; #` lista todos los grupos existentes

1.- Usando comandos de la API descrita anteriormente, escriba un *shell script* en *bash* llamado **grupo.sh** en el servidor *autentica*,

- a. que admita como entrada el nombre de un grupo, y cuya salida nos dé la descripción del grupo, el número de usuarios que contiene, y nos pregunte si queremos listarlos o no. En caso afirmativo, debe proporcionar la lista de usuarios pertenecientes al grupo.
- b. También debe comprobar al inicio de su ejecución si la entrada proporcionada es un grupo existente, y si no lo es, que cancele su ejecución, y proporcione como salida la información correspondiente.
- c. Además, si la capacidad ocupada en su unidad de red es mayor a un 75% debe informar de esto.

2.- Escriba:

- a. la línea necesaria en *crontab* para que el *script grupo.sh* se ejecute automáticamente cada noche a las 23:00, para conocer el estado del grupo "gerencia", y envíe su salida a un fichero localizado en */var/log/grupos/* llamado *status_gerencia*;
- b. el comando o comandos Linux que ejecutarían el *shell script grupo.sh* en el servidor *autentica*, desde otro servidor que se encuentra en la misma subred,
- c. y describa qué premisas son necesarias para que funcione correctamente esta llamada remota al script, y qué cuestiones de seguridad habría que tener en cuenta o afectan a esta ejecución remota.

3.- Los volúmenes definidos en el servicio de almacenamiento deben ser exportados vía NFS para que el servidor de copias de seguridad, *copiote.ipami.es*, se encargue de las copias. Describa la configuración necesaria para que los volúmenes puedan ser accedidos desde el servidor de copias de seguridad, e indique qué cuestiones de seguridad afectan a esta situación y cómo pueden ser configuradas correctamente.

FIN DEL SUPUESTO B

SUPUESTO C

La Universidad de Andalucía Oriental tiene delegado el dominio uniao.es. Para implementar el servicio disponemos de seis servidores DNS con BIND 9 con los siguientes roles:

- Un servidor primario autoritativo (primario).
- Dos servidores secundarios autoritativos (secundario1, secundario2).
- Una granja de tres servidores recursivos (recursivo1, recursivo2, recursivo3).

También se ofrece un servicio de correo electrónico con una topología de encaminamiento indirecto con los siguientes elementos:

- Granja de estafetas primarias que reciben y encaminan desde y hacia Internet los mensajes de correo de los dominios de mensajería que dispone la universidad (MTA) con Postfix.
- Granja de estafetas secundarias SMTP para dar servicio SMTPS y *submission* a los agentes de usuario y actuar como MDA mediante Postfix y dovecot.
- Granja IMAP para dar servicio IMAP los agentes de usuario utilizando dovecot.
- Una granja *webmail* como agente de usuario web usando RoundCube.

Para la autenticación y autorización se utiliza LDAP (openldap) siguiendo el modelo:

- Dos servidores maestros LDAP en configuración activo-activo donde se realizan las escrituras.
- Una granja de tres LDAP a la que acceden los clientes utilizando *dobles bind para autorizar, autenticar y provisión de atributos*.

Se recibe una incidencia de un usuario indicando que “tiene problemas con el correo”.

Basándonos en el diseño de arquitectura especificado anteriormente y los anexos 1 y 2:

- 1.- Indique qué pasos seguiría para diagnosticar la incidencia.
- 2.- Describa las posibles causas del “problema” y su solución a nivel de mensajería.
- 3.- Describa las posibles causas del “problema” y su solución a nivel de LDAP.

Proponga y justifique todas las suposiciones que estime oportunas.

Razone sus respuestas.

Anexo 1 DNS

Extracto de configuración de servidor primario autoritativo

named.conf

```
...
...
acl "trusted" {
    {
        151.204.25.1;
        localhost; };
};
acl "secundarios" {
    // DNS secundarios
    {
        151.204.60.1; // secundario1
        151.204.61.1; // secundario2
        151.204.62.1; // recursivo1
        151.204.63.1; // recursivo2
        151.204.64.1; // recursivo3
    };
};
acl "secundariosb" {
    // DNS secundarios
    {
        151.204.60.2; // secundario1b
        151.204.61.2; // secundario2b
        151.204.62.2; // recursivo1b
        151.204.63.2; // recursivo2b
        151.204.64.2; // recursivo3b
    };
};
acl "reduniao" {
    // Todas las subredes de uniao
    {
        151.204.25.0/24
        151.204.60.0/24
        151.204.61.0/24
        ...
        ...
        172.16.0.0/12;
        192.168.0.0/16;
        10.0.0.0/8;
    };
};
...
...
options {
    version none;
    notify yes;
    transfers-out 30;
    allow-transfer {
        "secundarios";
        "secundariosb";
    };
    allow-query { none; };
};
```

```

view "interna" {
match-clients { !secundarios; 151.204.25.1; secundariosb; !reduniao; };
allow-recursion { "trusted"; };
allow-query { 151.204.25.1;secundariosb; };
...
...
    zone "uniao.es" {
type master;
file "uniao.hosts.interno";
};
...
...
    zone "25.214.150.in-addr.arpa" {
type master;
file "uniao_25.rev";
};
...
...
    zone "1.168.192.in-addr.arpa" {
type master;
file "uniao_192.168.1.rev";
};
...
...
view "oficial" {
match-clients { !secundariosb; secundarios; !reduniao; any; };
recursion no;
...
...
    zone "uniao.es" {
type master;
file "uniao.hosts.externo";
allow-query { any; };
};
...
...
    zone "25.214.150.in-addr.arpa" {
type master;
file "uniao_25.rev";
};
};

```

uniao.hosts.externo

```
$TTL 7200
@ IN SOA dns0.uniao.es. hostmaster.uniao.es. (
2022031801 ; Serial
43200 ; refresh 12 horas
7200 ; retry 2 horas
2419200 ; expire 4 semanas
172800 ) ; minimum 2 dias
IN NS primario.uniao.es.
IN NS secundario1.uniao.es.
IN NS secundario2.uniao.es.
IN MX 10 estafetaprimaria.uniao.es.
IN TXT "v=spf1 ip4:151.204.200.1 ip4:151.204.200.2 ip4:151.204.200.3 -all"

primario IN A 151.204.25.1
secundario1 IN A 151.204.60.1
secundario2 IN A 151.204.61.1
$INCLUDE mapa.uniao.hosts.externo
```

Extracto de configuración de servidor recursivo

named.conf

```
...
...
acl "reduniao" { // Todas las subredes de uniao
{
151.204.25.0/24
151.204.60.0/24
151.204.61.0/24
...
...
172.16.0.0/12;
192.168.0.0/16;
10.0.0.0/8;
};
};
...
...
options {
...
...
allow-transfer { none; };
allow-query { "reduniao"; };
};
```

```
view "interna" {
match-clients { reduniao; };
transfer-source 151.204.60.2;
recursion yes;
...
...
zone "uniao.es" {
type slave;
file "uniao.hosts.interna.bck";
masters {
151.204.60.1;
};
};
// Resto de zonas internas de uniao
...
...
};

view "oficial" {
match-clients { any; };
transfer-source 151.204.60.2;
recursion no;
...
...
zone "uniao.es" {
type slave;
file "uniao.hosts.externa.bck";
allow-query { any; };
masters {
151.204.25.1;
};
};
// Resto de zonas externas de uniao
...
...
};
```

Anexo 2 LDAP

Extracto de configuración de servidor ldap

slapd.conf

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/nis.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema.propio/eduperson.schema
include /etc/openldap/schema.propio/schac.schema
include /etc/openldap/schema.propio/iris.schema
...
...
access to dn="cn=consulta,ou=perfil,dc=uniao,dc=es"
  by peername.ip="192.168.1.1" auth
  by peername.ip="192.168.1.2" auth
  ...
  ...
  by peername.ip="127.0.0.1" auth

access to
  filter="(eduPersonEntitlement=email)"
attrs=uid,uidNumber,gidNumber,homeDirectory,loginShell,gecos,cn,mail,shadowExpire
  by dn="cn=consulta,ou=perfil,dc=uniao,dc=es" read
  by * break
...
...
access to *
  by self read
  by users read
  by anonymous auth

### FIN slapd.conf
```

Consultas ldap

```
$ ldapsearch -L -D "cn=consulta,ou=perfil,dc=uniao,dc=es" -b
"ou=people,dc=uniao,dc=es" -H ldaps://ldap.uniao.es -W mail=nombre@uniao.es

# LDAPv3
# base <ou=people,dc=uniao,dc=es> with scope subtree
# filter: mail=usuario@uniao.es
# requesting: ALL
#

# usuario, people, uniao.es
dn: uid=usuario,ou=people,dc=uniao,dc=es
uid: usuario
mail: usuario@uniao.es
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
```

```

objectClass: eduPerson
objectClass: top
objectClass: shadowAccount
objectClass: irisPerson
uidNumber: 12000
gidNumber: 10000
gecos: Nombre Apellido1 Apellido2
loginShell: /usr/etc/ninguna
homeDirectory: /home/usuario
cn: NOMBRE APELLIDO1 APELLIDO2

$ ldapsearch -L -D "uid=usuario,ou=people,dc=uniao,dc=es" -b
"ou=people,dc=uniao,dc=es" -H ldaps://ldap.uniao.es -W mail=usuario@uniao.es
Enter LDAP Password:
version: 1

#
# LDAPv3
# base <ou=people,dc=uniao,dc=es> with scope subtree
# filter: mail=usuario@uniao.es
# requesting: ALL
#

# usuario, people, uniao.es
dn: uid=usuario,ou=people,dc=uniao,dc=es
uid: usuario
mail: usuario@uniao.es
objectClass : person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: eduPerson
objectClass: top
objectClass: shadowAccount
objectClass: irisPerson
uidNumber: 12000
gidNumber: 10000
givenName: NOMBRE
eduPersonEntitlement: email
eduPersonEntitlement: ftp
eduPersonAffiliation: staff
destinationIndicator: /home/usuario
shadowLastChange: 19843
gecos: Nombre Apellido1 Apellido2
loginShell: /usr/etc/ninguna
homeDirectory: /home/usuario
sn: APELLIDO1 APELLIDO2
cn: NOMBRE APELLIDO1 APELLIDO2

# search result

# numResponses: 2
# numEntries: 1

```

FIN DEL SUPUESTO C

SUPUESTO D

Imagine que forma parte del equipo de trabajo de Administración de Bases de Datos de una conocida universidad, y que las bases de datos que gestionan allí son Oracle 11 y 12. Se le encomienda lleve a cabo las siguientes tareas:

1.- Utilice el diccionario de datos para realizar las siguientes consultas:

1. Obtenga qué *datafiles* y *tablespaces* componen la base de datos.
2. Obtenga la localización de todas las copias de los online redo log file members, y su tamaño.
3. Obtenga el nombre y tamaño de el/los controlfile(s).
4. Obtenga el nombre y tamaño de los online redo log file members.
5. Obtenga el nombre y tamaño de los datafiles y tempfiles.

2.-Se le pide a continuación que escriba una serie de comandos para llevar a cabo las siguientes operaciones relacionadas con *tablespaces*:

1. Cree un *tablespace* llamado 'mitabspace' con un archivo de datos llamado 'tsfile01.dbf' que tenga un tamaño inicial de 10m. Utilice un tamaño de extensión uniforme de 64 Kb.
2. Cree dos tablas "tabla1" y "tabla2" con un campo de tipo fecha y otro numérico en el *tablespace* definido en el punto anterior.
3. Asigne manualmente algunas extensiones a cada una de ellas.
4. Ejecute una consulta para encontrar las ubicaciones físicas exactas del *tablespace* MITABSPACE (se pide consultar información sobre los segmentos, extensiones, bloques e identificadores de ficheros).
5. Cree el usuario NUEVO y dele los permisos necesarios para conectarse a la base de datos con autenticación por contraseña y poder crear tablas en su propio esquema. Haga que ese usuario trabaje por defecto en el *tablespace* 'mitabspace' sin límite de espacio.

3.-Por problemas de eficiencia, nos vemos obligados a particionar una tabla llamada VENTAS cuya definición es la siguiente:

```
CREATE TABLE VENTAS
( id_vendedor  NUMBER,
  id_cliente   NUMBER,
  id_item      NUMBER,
  region       VARCHAR(2) ,
  fecha        DATE
)
```

Nuestra intención es particionar la tabla en seis partes por los valores del campo *region* (columna con diecisiete valores posibles que van de 01 a 17).

Escriba el comando para crear una tabla llamada VENTAS_PARTICIONADA con el tipo de partición descrito.

Posteriormente, el administrador de la base de datos le indica que no ha quedado satisfecho con ese particionamiento en la tabla anterior y decide añadir una complejidad adicional. Concretamente, ahora deberá particionar la tabla en primer lugar por rango y posteriormente subparticionarla por los valores anteriores de regiones.

La primera partición será por rango atendiendo a los valores de la columna `id_item` (cuyos valores van de 1 a 10000) de forma que se harán cuatro partes: valores de `id_item` entre 1 y 2500, valores de `id_item` entre 2501 y 5000, valores de `id_item` entre 5001 y 7500 y el resto. Cada uno de estos rangos será a su vez subparticionado por el campo `región` de la misma forma descrita anteriormente, es decir, en seis partes atendiendo a los valores del campo `region` (columna con diecisiete valores posibles que van de 01 a 17).

Suponga que para las cuatro particiones por rango hay creados cuatro *tablespaces* llamados TS01, TS02, TS03 y TS04,

Escriba el comando para crear de nuevo la tabla VENTAS_PARTICIONADA con estas características.

FIN DEL SUPUESTO D

SUPUESTO E

La Universidad de Caladan ha puesto en marcha un proyecto para promocionar a su personal investigador. Una de las propuestas consiste en crear una página web en la que se muestren los investigadores e investigadoras más importantes de la Universidad basándose en el índice H.

El índice H se calcula ordenando las publicaciones de un investigador o investigadora por el número de citas recibidas en orden descendente y a continuación numerando e identificando el punto en el que el número de orden es mayor o igual que el número de citas recibidas por una publicación. Es decir, si tenemos un índice H = 7 será porque hay 7 publicaciones que han recibido al menos 7 citas cada una.

El personal investigador ha sido consultado con anterioridad y sólo se podrán publicar los datos de las personas que hayan dado su consentimiento.

Toda la información se ha recopilado y almacenado en una base de datos Oracle. Los parámetros de conexión a la base de datos son los siguientes:

SID	HOST	PUERTO	ESQUEMA	CLAVE
personal	atreides.caladan.es	1521	investigacion	sardaukar

Figura 1. Datos de conexión

La estructura de tablas que se ha definido la información se describe a continuación:

INVESTIGADORES		
<u>id_investigador</u>	varchar2(5), PK, not null	Identificador único del investigador/a
nombre	varchar2(50), not null	Nombre del investigador/a
apellido1	varchar2(50), not null	Primer apellido del investigador/a
apellido2	varchar2(50), not null	Segundo apellido del investigador/a
departamento	varchar2(100)	Departamento al que se encuentra afiliado
consiente_publicacion	varchar2(1), not null, check('S','N'), default='N'	Indica si el investigador/a ha dado el consentimiento para la publicación de sus datos. S en caso de que la haya dado, N en caso de que no. N será el valor por defecto

ARTICULOS		
<u>id_articulo</u>	varchar2(5), PK, not null	Identificador único del artículo
titulo	varchar2(100), not null	Título del artículo
resumen	clob	Resumen del artículo
fecha_publicacion	date	Fecha de publicación del artículo
id_tipo_publicacion	varchar2(3), FK(tipo_publicacion.id_tipo_publicacion)	Tipo de publicación

CONTRIBUIDORES		
<u>id_investigador</u>	varchar2(5), PK, FK(investigadores.id_investigador), not null	Identificador único del investigador. Hace referencia al campo id_investigador de la tabla investigadores
<u>id_articulo</u>	varchar2(5), PK, FK(articulos.id_articulo), not null	Identificador único del artículo. Hace referencia al campo id_articulo de la tabla articulo

BIBLIOGRAFIA		
<u>id_articulo</u>	varchar2(5), PK, FK(articulos.id_articulo), not null	Identificador único del artículo
<u>id_articulo_citado</u>	varchar2(5), PK, FK(articulos.id_articulo), not null	Identificador único del artículo al que se cita

TIPO_PUBLICACION		
id_tipo_publicacion	varchar2(3), PK, not null	Identificador único del tipo de publicación
desc_tipo_publicacion	varchar2(50), not null	Descripción del tipo de publicación

1.- Implemente en Java las clases y métodos necesarios para recuperar de la base de datos la información de cada investigador o investigadora. Escriba el código necesario que realice la conexión a la base de datos descrita en la tabla *“datos de conexión”*. Incluya la consulta SQL que obtenga sus datos, así como el número de citas de cada una de sus publicaciones. Tenga en cuenta las estructuras de tablas descritas anteriormente.

2.- A partir de los datos obtenidos en el apartado anterior se le pide implementar un algoritmo en Java, incluyendo las clases y métodos necesarios, que permita calcular el índice H de cada investigador o investigadora.

3.- En la misma página web se quiere proporcionar a los usuarios un buscador de personal investigador. Implemente los siguientes elementos:

- Una interfaz de búsqueda vía web que debe solicitar los resultados al servidor y mostrarlos sin recargar la página.
- El código necesario para definir una API REST que acepte la búsqueda de peticiones, y que devuelva los resultados en JSON.

FIN DEL SUPUESTO E

SUPUESTO F

Cierta universidad española dispone de una base de datos en Oracle 11 que se utiliza para gestionar el Personal Docente e Investigador de dicha universidad. Para este supuesto utilizaremos las siguientes tablas: PERSONAL, CONTRATOS, PLUSES, CATEGORIAS, CENTROS, DEPARTAMENTOS que se describen a continuación:

-Datos personales del PDI. Campos: DNI, apellidos y nombre, fecha de nacimiento y sexo (valores H o M).

PERSONAL (

DNI VARCHAR2(9) NOT NULL,
APELLIDO1 VARCHAR2(20) NOT NULL,
APELLIDO2 VARCHAR2(20),
NOMBRE VARCHAR2(20) NOT NULL,
FECNACI DATE NOT NULL,
SEXO VARCHAR2(1) NOT NULL);

Clave primaria de PERSONAL: DNI

check (SEXO in ('H', 'M'))

Campo sexo: valores H y M correspondientes a Hombre y Mujer, respectivamente

-Contratos del PDI. Campos DNI, fechas de inicio y fin del periodo de contrato (si la fecha de fin es nula el contrato está en vigor), código de categoría del contrato, código del centro y código del departamento en el que está contratado el PDI

CONTRATOS (

DNI VARCHAR2(9) NOT NULL,
FECINI DATE NOT NULL,
FECFIN DATE,
CATEGORIA VARCHAR2(5) NOT NULL,
CENTRO VARCHAR2(5) NOT NULL,
DEPARTAMENTO VARCHAR2(5) NOT NULL);

Clave primaria de CONTRATOS: DNI, FECINI

foreign key (DNI) references PERSONAL(DNI)

foreign key (CATEGORIA) references CATEGORIAS(CODIGO)

foreign key (CENTRO) references CENTROS(CODIGO)

foreign key (DEPARTAMENTO) references DEPARTAMENTOS(CODIGO)

-Pluses obtenidos por el PDI. Campos DNI, fechas de inicio y fin del periodo en el que se cumple el plus, código del plus obtenido (TR trienios, TD tramos docente, TI tramos de investigación), número de pluses obtenidos de ese tipo en ese periodo.

PLUSES (

DNI VARCHAR2(9) NOT NULL,
FECINI DATE NOT NULL,
FECFIN DATE NOT NULL,
PLUS VARCHAR2(5) NOT NULL,
NUMERO NUMBER(2) NOT NULL);

Clave primaria de PLUSES: DNI, FECINI, PLUS

foreign key (DNI) references PERSONAL(DNI)

check (PLUS in ('TR','TD','TI'))

Campo plus: valores TR, TD, TI correspondientes a trienios, tramos docentes y tramos de investigación, respectivamente.

-Tablas de códigos de categorías, códigos de centros y códigos de departamentos. Campos CODIGO y NOMBRE en los tres casos.

CATEGORIAS (

CODIGO VARCHAR2(5) NOT NULL,

NOMBRE VARCHAR2(100) NOT NULL);

Clave primaria de CATEGORIAS: CODIGO

CENTROS (

CODIGO VARCHAR2(5) NOT NULL,

NOMBRE VARCHAR2(100) NOT NULL);

Clave primaria de CENTROS: CODIGO

DEPARTAMENTOS (

CODIGO VARCHAR2(5) NOT NULL,

NOMBRE VARCHAR2(100) NOT NULL);

Clave primaria de DEPARTAMENTOS: CODIGO

-Vista del personal con sus edades. Vista EDADES con los campos DNI, sexo (valores H y M correspondientes a Hombre y Mujer, respectivamente) y edad.

La vista original se crea de la siguiente forma:

```
CREATE OR REPLACE VIEW EDADES AS
  SELECT p.dni,
         p.sexo,
         TRUNC (MONTHS_BETWEEN (SYSDATE, p.fecnaci) / 12) EDAD
  FROM PERSONAL p, CONTRATOS c
 WHERE   p.dni = c.dni
        AND c.fecfin IS NULL;
```

La vista EDADES contiene el personal activo (aquel que tiene un contrato con la fecha de fin abierta) indicando DNI, sexo (H o M), y edad. Una muestra de esta vista es:

<u>DNI</u>	<u>SEXO</u>	<u>EDAD</u>
DNI1	H	25
DNI2	M	37
DNI3	H	48
DNI4	M	61
DNI5	H	50
DNI6	M	59

Con toda esta información, se pide:

1.- Modificar la vista anterior EDADES para añadirle un campo nuevo que distinga la franja de edad por decenas (por ejemplo, para aquellos que tienen una edad entre 20-29 el campo mostraría 20, aquellos que tienen una edad entre 30-39, el campo mostraría 30, etc.)

Ejemplo:

<u>DNI</u>	<u>SEXO</u>	<u>EDAD</u>	<u>FRANJA</u>
DNI1	H	25	20
DNI2	M	37	30
DNI3	H	48	40
DNI4	M	61	60
DNI5	H	50	50
DNI6	M	59	50

Una vez modificada la vista con esa información, cree una vista transpuesta llamada ESTADISTICA_EDADES que muestre información estadística por sexo y edades de la siguiente forma:

<u>SEXO</u>	<u>20-29</u>	<u>30-39</u>	<u>40-49</u>	<u>50-59</u>	<u>60-69</u>	<u>70-79</u>
Hombre	99	332	709	222	121	45
Mujer	88	279	1001	555	767	56

2.- Crear una vista llamada DNI_PLUSES con los pluses del personal activo (columna FECFIN vacía) que contenga la siguiente información: DNI, APELLIDOS Y NOMBRE, NOMBRE DEL CENTRO, NOMBRE DEL DEPARTAMENTO, NOMBRE DE LA CATEGORIA, NÚMERO DE TRIENIOS, NÚMERO DE TRAMOS DOCENTES, NÚMERO DE TRAMOS DE INVESTIGACIÓN.

El resultado de la vista debe ser similar al siguiente:

<u>DNI</u>	<u>APENOM</u>	<u>CENTRO</u>	<u>DPTO</u>	<u>CATEG</u>	<u>NUMTRI</u>	<u>NUMTRADOC</u>	<u>NUMTRAINV</u>
...							
DNI1	NOMBRE1	FACUL.1	DEP.1	CAT.1	5	3	1
DNI2	NOMBRE2	FACUL.2	DEP.2	CAT.2	7	4	2
DNI3	NOMBRE3	FACUL.3	DEP.3	CAT.3	9	5	3
DNI4	NOMBRE4	FACUL.4	DEP.4	CAT.4	2	1	0
...							

3.- Suponga que existe un form que invoca a otro form llamado **'control'** enviando en esa llamada una serie de parámetros PDNI, PESQUEMA, PBASE, que se corresponden con el DNI de la persona que quiere acceder a la aplicación, el esquema y la base de datos a la que se conectará la aplicación:

El código que invoca al form **'control'** es:

```
(...)  
pl := Get_Parameter_List('tmp');  
IF not Id_Null(pl) THEN  
  Destroy_Parameter_List(pl);  
END IF;  
pl := Create_Parameter_List('tmp');  
Add_Parameter(pl, 'PDNI', TEXT_PARAMETER, '11223344');  
Add_Parameter(pl, 'PESQUEMA', TEXT_PARAMETER, 'PERSONAL');  
Add_Parameter(pl, 'PBASE', TEXT_PARAMETER, 'mibase');  
run_product('control', Asynchronous, runtime, filesystem, pl);  
(...)
```

Suponga que tenemos definida una función PLSQL accesible desde el form **'control'** definida como:

```
FUNCTION conecta (pesquema in varchar2,  
                 pbase in varchar2) return number
```

Esta función sirve para realizar la conexión a un esquema/base de bases de datos. Devuelve 1 si la conexión ha sido correcta y 0 en caso contrario.

Queremos que el formulario **'control'** se conecte automáticamente al esquema de bases de datos enviado por parámetro.

En caso de que la conexión sea incorrecta se mostrará un mensaje ("**Conexión incorrecta**") y se saldrá del form.

En caso contrario, si la conexión ha resultado bien, se comprobará que el acceso al formulario **'control'** se está realizando por un usuario autorizado. Para ello hay que comprobar que el DNI recibido como parámetro esté como usuario activo en la siguiente tabla:

```
USUARIOS (DNI VARCHAR2(9) NOT NULL,  
          ACTIVO VARCHAR2(1) NOT NULL);
```

El campo "activo" puede tomar dos valores 'S' si es un usuario activo y 'N' si no está activo,

Si al comprobar resultara que el DNI no existe en dicha tabla como usuario activo se mostrará un mensaje ("**Usuario no autorizado**") y se saldrá del form.

Explique qué elementos tendría que definir en el form **'control'** y programe el/los triggers que considere necesarios para cumplir los requisitos anteriores.

FIN DEL SUPUESTO F