



## EJERCICIO 1 – BASES DE DATOS (2.5 puntos)

El esquema lógico relacional, abajo indicado, corresponde a una empresa que se dedica a la venta y distribución de libros a determinadas librerías. Esta empresa, compra los libros directamente a las editoriales, y los ejemplares se guardan en distintos almacenes que tiene repartidos por la provincia. Cuando llega un pedido de una librería, se sirve en su totalidad, si esto es posible, o se queda pendiente hasta que haya existencias suficientes en los almacenes. El pedido, realizado por la librería, es atendido en uno de los almacenes. Si hay existencias suficientes para suministrar todos los libros del pedido, es servido por uno de los vendedores que la distribuidora tiene en plantilla, minorándose las existencias en los almacenes que han aportado ejemplares.

### Esquema relacional

**EDITORIAL** (COD\_EDIT: STRING, NOM\_EDIT: STRING, DIR\_EDIT: STRING, CIU\_EDIT: STRING)

CP: {COD\_EDIT}

VNN: {NOM\_EDIT}

**ALMACEN** (COD\_ALM: STRING, NOM\_ALM: STRING, DIR\_ALM: STRING, CIU\_ALM: STRING, TELEF\_ALM: STRING),

CP: {COD\_ALM}

VNN: {NOM\_ALM}

UNIQUE: {NOM\_ALM}

**LIBRO** (NUM\_REF: STRING, TITULO: STRING, AUTORES: STRING, PRECIO\_COM: REAL, PRECIO\_VEN: REAL, EDITOR: STRING)

CP: {NUM\_REF}

CAJ: {EDITOR} -> EDITORIAL (COD\_EDIT)

VNN: {TITULO}

VNN: {AUTORES}

VNN: {EDITOR}

CHECK: {PRECIO\_VEN > PRECIO\_COM}

**LIBRERIA** (COD\_LIB: STRING, NOM\_LIB: STRING, DIR\_LIB: STRING, CIU\_LIB: STRING, TELEF\_LIB: STRING)

CP: {COD\_LIB}

VNN: {NOM\_LIB}

VNN: {TELEF\_LIB}

**VENDEDOR** (COD\_VEN: STRING, NOM\_VEN: STRING, TELEF\_VEN: STRING, COMISION: REAL)

CP: {COD\_VEN}

CHECK: {COMISION < 20}

**EXISTE** (ALMAC: STRING, REFER: STRING, CANTIDAD: ENTERO, CANT\_MIN: ENTERO)

CP: {ALMAC, REFER}

CAJ: {ALMAC} -> ALMACEN(COD\_ALM)

CAJ: {REFER} -> LIBRO (NUM\_REF)



VNN: {CANTIDAD}  
VNN: {CANT\_MIN}  
CHECK: {CANTIDAD>0}  
CHECK: {CANT\_MIN>0}

**PEDIDO** (NUM\_PED: ENTERO, LIBRER: STRING, ALMAC: STRING, VENDED: STRING, FECHA: DATE, PENDIENTE: LÓGICO)

CP: {NUM\_PED}  
CAJ: {LIBRER} -> LIBRERÍA (COD\_LIB)  
CAJ: {ALMAC} -> ALMACEN (COD\_ALM)  
CAJ: {VENDED} -> VENDEDOR (COD\_VEN)  
VNN: {FECHA}  
CHECK: {CANT\_PED>0}

**LINEA** (NUM\_PED: ENTERO, NUM\_REF: STRING, CANT\_PED: ENTERO)

CP: {NUM\_PED, NUM\_REF}  
CAJ: {NUM\_PED} -> PEDIDO(NUM\_PED)  
CAJ: {NUM\_REF} -> LIBRO(NUM\_REF)  
VNN: {CANT\_PED}  
CHECK: {CANT\_PED>0}

## NOTAS

- En el esquema relacional se representan las siguientes restricciones:
  - CP: Clave primaria
  - CAJ: Clave ajena o foránea. En ella se indica la tabla referenciada y su campo en dicha tabla.
  - UNIQUE: Clave alternativa
  - VNN: Restricción de valor no nulo
  - Restricciones de dominio
    - CHECK: restricción de comprobación
    - Restricciones de tipo: ENTERO (int, integer, long ...), REAL (float, double ...), STRING (cadena de caracteres), DATE (tipo fecha) Y LÓGICO (booleano)
- EDITORIAL: Contiene datos de las editoriales donde se compran los libros
- LIBRO: Contiene datos de los libros y su editorial
- ALMACEN: Datos de los almacenes donde la empresa guarda los libros
- EXISTE: Contiene la relación entre libro y almacén junto a la cantidad disponible. El campo cantidad mínima marca el nivel de reposición y por tanto de solicitud a las editoriales de nuevos ejemplares.
- LIBRERIA: Almacena datos de las librerías donde se distribuyen los libros.
- VENDEDOR: Contiene los datos personales del vendedor, incluido su comisión
- PEDIDO: Se almacenan datos de los pedidos, almacén que lo atiende, librería que hace el pedido, vendedor que lo distribuye, fecha del pedido e información de si el pedido está pendiente de ser servido completamente o no.
- LINEA: Contiene las líneas de los pedidos, así como la cantidad de ejemplares de ese libro pedido.



**Ejercicio 1.1 – (0.5 puntos)**

Realiza una consulta que muestre el título, autores y nombre de la editorial de aquellos libros para los que se hayan pedido más de 10 ejemplares.

**Ejercicio 1.2 – (0.5 puntos)**

Define una vista de forma que, para cada libro que aparezca en un pedido no servido, muestre el número de referencia, título autores, nombre de la editorial y existencias en todos los almacenes.

**Ejercicio 1.3 – (0.75 puntos)**

Realizar un procedimiento almacenado que permita incrementar en dos puntos la comisión de los tres vendedores que hayan servido más pedidos. Ejemplo de llamada: `incr_comision()`

**Ejercicio 1.4 – (0.75 puntos)**

Realizar un trigger que compruebe que no puede aumentarse la comisión a un vendedor cuando el porcentaje de pedidos servidos por él caiga por debajo del 20% del total de pedidos servidos. Revocar la operación en caso que se produzca tal suceso.



## EJERCICIO 2 – LENGUAJES DE MARCAS (2.5 puntos)

A partir del archivo almacen.xml, cuyo contenido es el siguiente, responde a las cuestiones planteadas.

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="almacen.xsl"?>
<almacen xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="almacen.xsd">
<almacen>
  <paquete codigo="AL001">
    <peso unidad_medida="oz">31</peso>
    <Volumen_mediano/>
    <situacion zona="229">
      <estanteria>5</estanteria>
      <balda>3</balda>
    </situacion>
  </paquete>
  <paquete codigo="bZ001">
    <peso unidad_medida="g">1193.50</peso>
    <Volumen_pequeño/>
    <situacion zona="229">
      <estanteria>5</estanteria>
      <balda>4</balda>
    </situacion>
    <fragil/>
  </paquete>
  <paquete codigo="tr100">
    <peso unidad_medida="oz">7193</peso>
    <Volumen_grande/>
    <situacion zona="004">
      <estanteria>3</estanteria>
      <balda>1</balda>
    </situacion>
  </paquete>
  <paquete codigo="CC075">
    <peso unidad_medida="g">40192.50</peso>
    <Volumen_grande/>
    <situacion zona="201">
      <balda>5</balda>
      <estanteria>1</estanteria>
    </situacion>
    <fragil/>
  </paquete>
  <paquete codigo="Pw907">
    <peso unidad_medida="g">75</peso>
    <Volumen_pequeño/>
    <situacion zona="004">
      <estanteria>1</estanteria>
      <balda>7</balda>
    </situacion>
  </paquete>
</almacen>

```



### Ejercicio 2.1 – (1.75 puntos)

- A) Utilizando el documento `almacen.xml`, crear el fichero `almacen.xsd` (XML Schema Definition) que valide el xml dado.
- B) Además, que cumpla las siguientes condiciones:
- Existirá un único elemento vacío de entre estos tres: `Volumen_pequeño`, `Volumen_mediano`, `Volumen_grande`
  - El código del paquete, es obligatorio y está compuesto por dos letras, mayúsculas o minúsculas y tres dígitos.

```
<?xml version="1.0" encoding="UTF-8"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">  
  <xs:element name="almacen" type="T_almacen"/>  
</xs:schema>
```

Respuesta pregunta 1, crea el tipo `T_almacen` correspondiente y todos los necesarios

```
</xs:schema>
```

### Ejercicio 2.2 – (0.75 puntos)

A partir del fichero `almacen.xml`, escribe una consulta XQuery que devuelva:

- El total del peso de los paquetes cuya unidad de medida sea "g" (gramos).
- El total del peso de los paquetes cuya unidad de medida sea "oz" (onzas).
- (Si aproximadamente cada onza son unos 28 gramos). El total del peso de los paquetes cuya unidad de medida sea "oz" (onzas) convertido a gramos.
- El peso total de todos los paquetes expresado en gramos.

Salida de la consulta XQuery pedida:

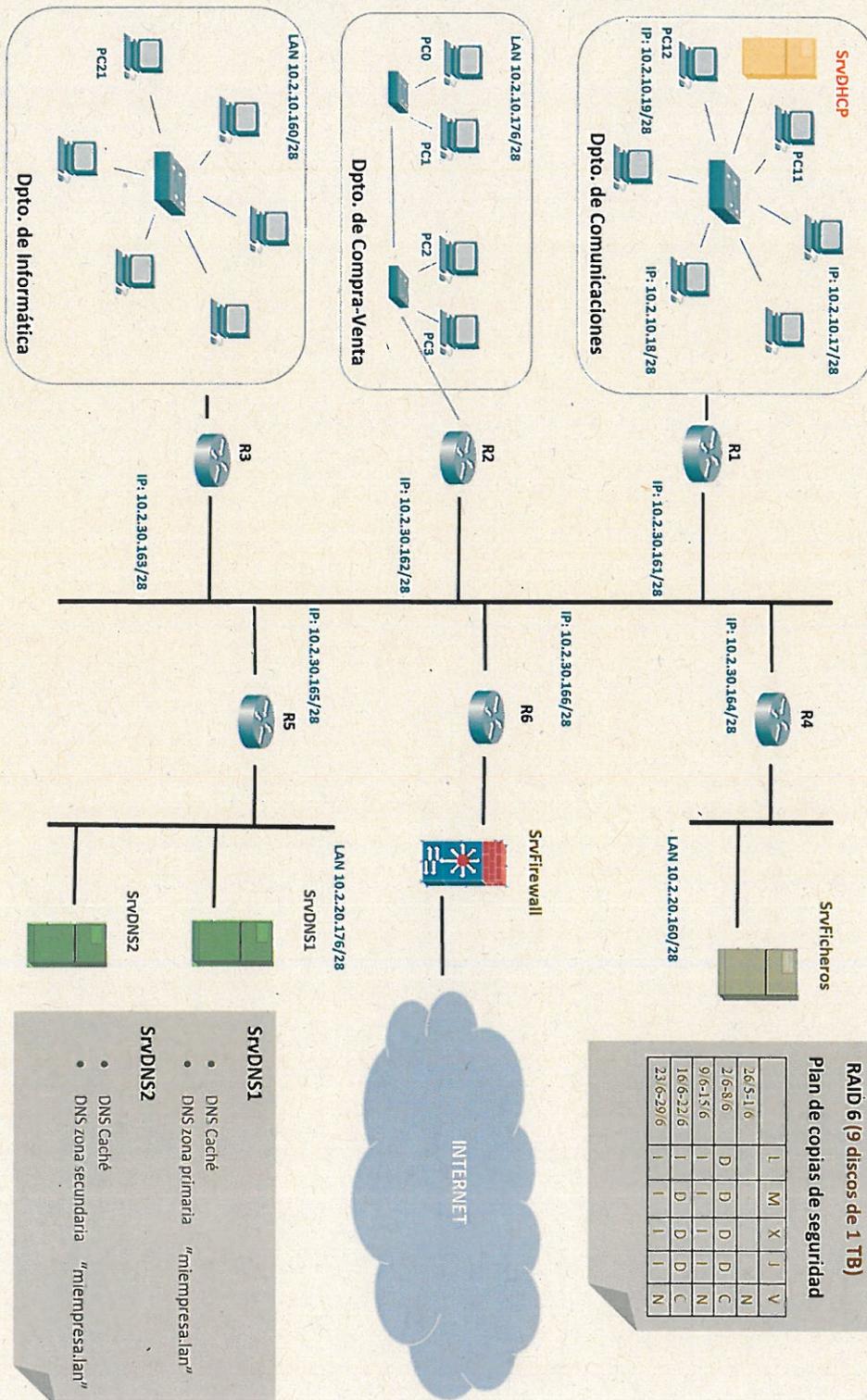
```
<total_gramos>41461</total_gramos>  
<total_onzas>7224</total_onzas>  
<total_onzas_en_gramos>202272</total_onzas_en_gramos>  
<total_peso_en_gramos>243733</total_peso_en_gramos>
```

Respuesta pregunta 2



### EJERCICIO 3 – REDES Y SERVICIOS (2.5 puntos)

Teniendo en cuenta el siguiente esquema responde a las preguntas planteadas.





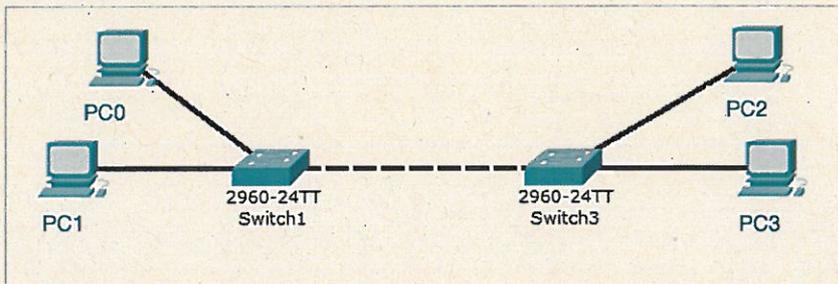
**Ejercicio 3.1 – (0.3 puntos)**

Suponiendo que dejamos para los routers R1, R2 y R3 las IPs más altas de las subredes a las que pertenecen, indica la IP que tendría cada uno de ellos en las subredes de los departamentos:

Dpto.	Router	IP
Comunicaciones	R1	
Compra-Venta	R2	
Informática	R3	

**Ejercicio 3.2 – (0.1 puntos)**

En la siguiente figura se muestra la LAN del Dpto. de Compra-Venta. Está formada por 4 PCs de sobremesa conectados a dos switches que a su vez están conectados entre sí. ¿Cuántos dominios de colisión tenemos en esta figura?



**Ejercicio 3.3 – (0.1 puntos)**

Se decide unir el Switch3 con el R2 mediante cableado UTP categoría 6 ¿Cuál sería la distancia máxima garantizada para este tipo de cableado?

**Ejercicio 3.4 – (0.3 puntos)**

Se está pensando en dividir la subred 10.2.10.176/28 del Dpto. de Compra-Venta en dos subredes con el máximo número de hosts cada una. Para cada una de las dos subredes indica el número máximo de hosts, la dirección de subred, la dirección de broadcast y la máscara

Subred	Nº máx. Hosts	Máscara	Dirección Subred	Dirección Broadcast
Compra				
Venta				



**Ejercicio 3.5 – (0.1 puntos)**

El PC1 le manda un mensaje al PC3. ¿En qué capa del modelo OSI se han añadido los datos a la PDU enviada durante el proceso de encapsulación?

**Ejercicio 3.6 – (0.1 puntos)**

¿Qué capas del modelo OSI tiene como mínimo el Switch1 del Dpto. de Compra-Venta independientemente del tipo de switch que sea?

**Ejercicio 3.7 – (0.1 puntos)**

¿Cuál es la dirección de **loopback** de IPv6?

**Ejercicio 3.8 – (0.1 puntos)**

Un usuario del PC1 del Dpto. de Compra-Venta ha borrado sin querer los ficheros que había preparado para un cliente. Dichos ficheros no están en la papelera. Cita una herramienta gratuita que se utilice en la actualidad y que permita **recuperar ficheros borrados** recientemente.

**Ejercicio 3.9 – (0.2 puntos)**

Una vez que ya le funciona Internet sin problemas, el usuario del PC1 teclea la siguiente dirección en el navegador: <https://www.google.es>

- a) ¿Qué parte de la dirección es el **FQDN**?
- b) ¿Qué parte es el **host**?

**Ejercicio 3.10 – (0.2 puntos)**

Imaginemos la tabla de enrutamiento de PC1.

- a) ¿Cuál sería la **máscara de enrutamiento** IPv4 del host de destino 10.2.10.50 en dicha tabla de enrutamiento?
- b) ¿Y la **máscara de enrutamiento** de la puerta de enlace predeterminada?

**Ejercicio 3.11 – (0.2 puntos)**

Obtener la/s superred/es que **sumarice** en lo posible las 3 LAN de los departamentos de tal forma que no incluya subredes que no tuviéramos antes.


**Ejercicio 3.12 – (0.1 puntos)**

En el Servidor de Ficheros SrvFicheros tenemos configurado un RAID 5 con 9 discos de 1 TB cada uno. ¿De qué espacio neto disponemos para datos?

**Ejercicio 3.13 – (0.3 puntos)**

En ese mismo servidor de ficheros tenemos configurado el siguiente plan de copias de seguridad:

	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
26/5-1/6					Normal 1
2/6-8/6	Diferencial 1	Diferencial 2	Diferencial 3	Diferencial 4	Copia 1
9/6-15/6	Incremental 1	Incremental 2	Incremental 3	Incremental 4	Normal 2
16/6-22/6	Diferencial 5	Diferencial 6	Diferencial 7	Diferencial 8	Copia 2
23/6-29/6	Incremental 5	Incremental 6	Incremental 7	Incremental 8	Normal 3

- El viernes 13 de junio perdemos todos los datos **antes de realizar la Normal 2**. ¿Qué tendremos que restaurar para recuperarlo todo? (0.2 puntos)
- ¿Qué diferencia hay entre hacer una copia de seguridad de tipo **Normal** o de tipo **Copia** los viernes? (0.1 puntos)

**Ejercicio 3.14 – (0.1 puntos)**

Intentamos crear manualmente el registro **PC1.miempresa.ian** asociado a la IP 10.2.5.129 en la zona "miempresa.ian" del SrvDNS2 pero nos da un mensaje de error y no podemos crearlo. ¿A qué es debido?

**Ejercicio 3.15 – (0.2 puntos)**

¿Qué es un Relé de DHCP y cuándo se utiliza?



## EJERCICIO 4 – PROGRAMACIÓN (2.5 puntos)

El objetivo de este ejercicio es crear un programa en Java para generar contraseñas. Dicho programa cumplirá las siguientes condiciones:

1. Se crearán tres arrays, en la función principal, con los siguientes contenidos:
  - a. Un array que contenga las letras minúsculas (a - z) y mayúsculas (A - Z).
  - b. Un array con los números (0 - 9).
  - c. Un array con caracteres especiales (diez caracteres).
2. El programa ofrecerá un menú que ofrecerá las siguientes opciones:
  - a. Imprimir el contenido de los arrays. Los elementos de los array se imprimirán separados por "comas" y después del último elemento aparecerá un "punto".
  - b. Generar una contraseña débil compuesta por 5 letras y 3 números. Se utilizarán los arrays definidos en el programa principal. Tanto el elemento, como el tipo de elemento, se obtendrán de forma aleatoria.
  - c. Generar una contraseña fuerte compuesta por 6 letras, 4 números y 3 caracteres especiales. Se utilizarán los arrays definidos en el programa principal. Tanto el elemento como el tipo de elemento se obtendrán de forma aleatoria. En la primera posición aparecerá un carácter especial, la primera letra generada debe ser obligatoriamente mayúscula, el resto minúsculas y ningún elemento se podrá repetir.
  - d. Ofrecer la salida del programa.

Todas las opciones del menú (excepto salir) se realizarán con una función a la que se le pasarán como parámetros los arrays.

### Ayuda:

Declarar un array con caracteres numéricos :

```
char [] numeros= {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};
```

Número aleatorio entre 0 y 9: (int) (Math.random()\*10);

### Puntuación detallada del ejercicio:

0.5 puntos por:

- La creación del menú.
- Las llamadas a funciones.
- La función para imprimir los arrays.

0.75 puntos por:

- La función para generar la contraseña débil.

1.25 puntos por:

- La función para generar la contraseña fuerte.

### Ejemplo:

```
----- Menú -----
1. Imprimir arrays.
2. Genera e imprime un password débil.
3. Genera e imprime un password fuerte.
4. Salir.
1
Números:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
Letras:
a, b, c, d, e, f, g, h, i, j, A, B, C, D, E, F, G, H, I, J.
Caracteres especiales :
!, @, #, $, %, &, /, (, ).
----- Menú -----
1. Imprimir arrays.
2. Genera e imprime un password débil.
3. Genera e imprime un password fuerte.
4. Salir.
2
Password Débil:
Da38G4bH
----- Menú -----
1. Imprimir arrays.
2. Genera e imprime un password débil.
3. Genera e imprime un password fuerte.
4. Salir.
3
Password Fuerte:
/E@4fb$e79jc8
----- Menú -----
1. Imprimir arrays.
2. Genera e imprime un password débil.
3. Genera e imprime un password fuerte.
4. Salir.
4
Gracias. Vuelve pronto
```