

Orden EDU/1406/2023, de 4 de diciembre (BOCyL de 12 de diciembre)

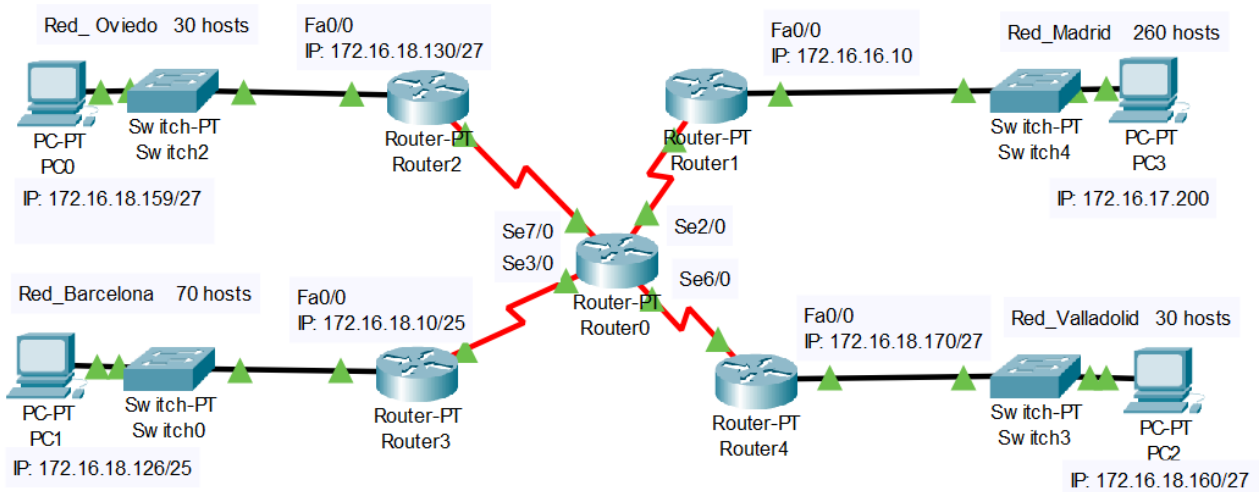
CUERPO:	PROFESORES DE ENSEÑANZA SECUNDARIA
ESPECIALIDAD:	INFORMÁTICA
PRUEBA:	PRIMERA PARTE (PRÁCTICO) DE LA PRIMERA PRUEBA
TURNO:	1 y 2

BLOQUE I REDES (3 puntos)

(Este bloque se contestará en estas mismas hojas de examen)

EJERCICIO 1.1 (1,75 puntos)

Partiendo de la red **172.16.16.0/21** se realiza un Direccionamiento de máscaras de longitud variable **VLSM**. Observar que, además de las redes de paso, la *Red Oviedo* necesita 30 hosts, la *Red Barcelona* necesita 70 hosts, la *Red Valladolid* necesita 30 host y la *Red Madrid* necesita 260 host.



a) Indicar la máscara que debe tener la **Red Madrid**

	Máscara en Formato decimal	Máscara en Formato CIDR
Red Madrid		

Orden EDU/1406/2023, de 4 de diciembre (BOCyL de 12 de diciembre)

b) Configurar las interfaces del Router 0:

Dispositivo	Interfaz	IP	Máscara en Formato CIDR
Router 0	Se2/0		
	Se7/0		
	Se3/0		
	Se6/0		

c) Indicar si hay errores en la configuración de IPs dadas en la topología que no permita la comunicación.

Dispositivo/Interface	Configuración correcta (IP/Máscara en formato CIDR)	Descripción del error

Orden EDU/1406/2023, de 4 de diciembre (BOCyL de 12 de diciembre)

EJERCICIO 1.2 (0,50 puntos)

Realiza la **sumarización** de rutas estáticas de la siguiente tabla de enrutamiento.

Router

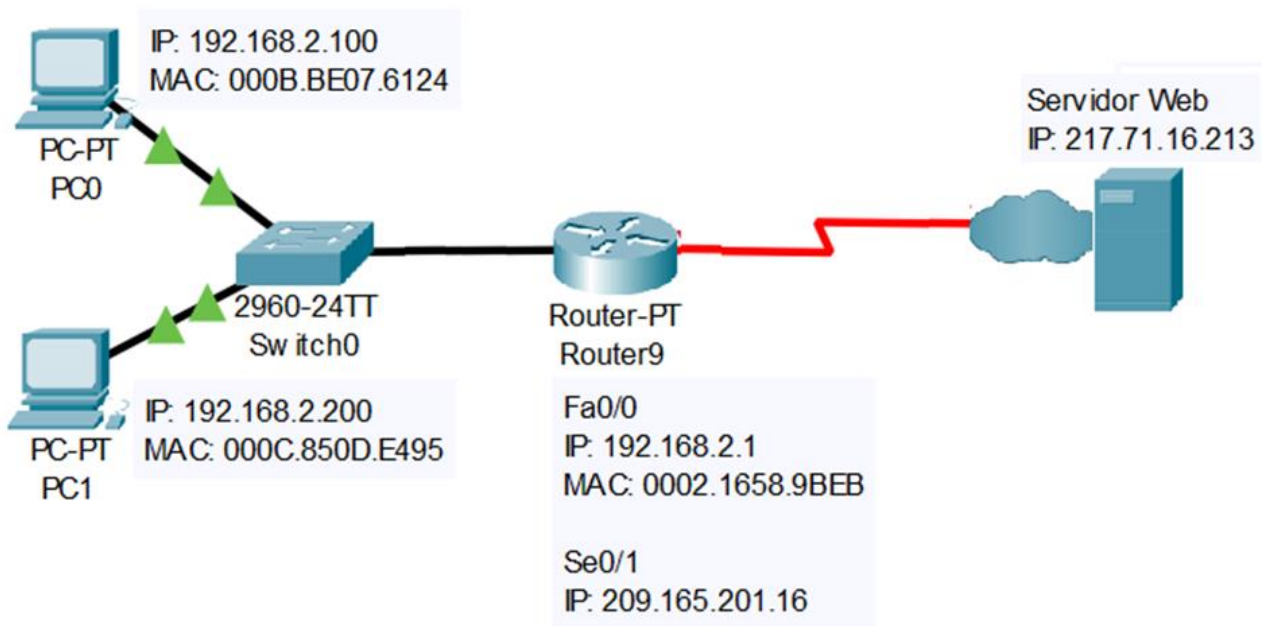
Tipo	Red Destino	Máscara de Subred	Puerta de enlace	Interfaz de salida	Métrica
C	10.10.0.0	255.255.0.0	10.10.0.2	10.10.0.2	1
C	10.20.0.0	255.255.0.0	10.20.0.2	10.20.0.2	1
C	10.30.0.0	255.255.0.0	10.30.0.2	10.30.0.2	1
S	172.16.1.0	255.255.255.0	10.30.0.1	10.30.0.2	2
S	172.16.2.0	255.255.255.0	10.30.0.1	10.30.0.2	2
S	172.16.80.0	255.255.240.0	10.10.0.1	10.10.0.2	2
S	192.168.10.64	255.255.255.224	10.20.0.1	10.20.0.2	2
S	172.16.96.0	255.255.240.0	10.10.0.1	10.10.0.2	2
S	172.16.3.0	255.255.255.0	10.30.0.1	10.30.0.2	2

Tipo	Red Destino	Máscara de Subred	Puerta de enlace	Interfaz de salida	Métrica
C	10.10.0.0	255.255.0.0	10.10.0.2	10.10.0.2	1
C	10.20.0.0	255.255.0.0	10.20.0.2	10.20.0.2	1
C	10.30.0.0	255.255.0.0	10.30.0.2	10.30.0.2	1

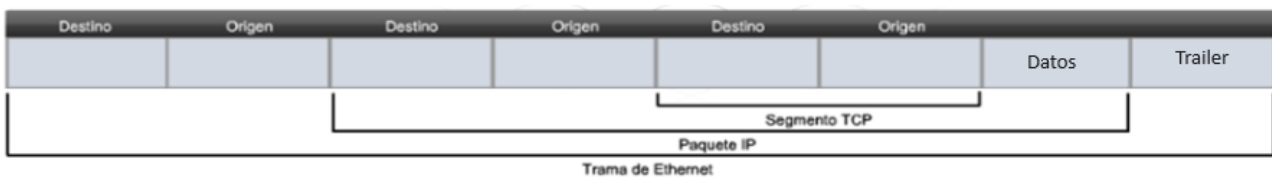
Orden EDU/1406/2023, de 4 de diciembre (BOCyL de 12 de diciembre)

EJERCICIO 1.3 (0,75 puntos)

Trama Ethernet. PC0 quiere consultar la página del Servidor Web.



Trama Ethernet que sale de PC0. Escribir los datos sobre la trama.



¿En caso de no tenerla, con qué protocolo consigue PC0 la MAC Destino de la trama?

¿En caso de no tenerla, con qué protocolo consigue PC0 la IP de la Pág. web?

Orden EDU/1406/2023, de 4 de diciembre (BOCyL de 12 de diciembre)

BLOQUE II SISTEMAS (1 punto)

(Este bloque se contestará en estas mismas hojas de examen)

EJERCICIO 2.1 (0,5 Puntos)

En una red se da la siguiente situación:

- Existe una máquina con ip 10.0.0.1 , que dispone de un servicio openssh accesible en el puerto 55, y donde está creado un usuario llamado *usrRemoto*
- Existe una maquina con ip 10.0.0.2 , que dispone de software cliente openssh y donde está creado un usuario llamado *usrLocal*
- No existen cortafuegos y ambas maquinas responden al ping entre ellas.
- Se desea que el usuario *usrLocal* pueda abrir una sesión ssh en la maquina 10.0.0.1, desde la máquina 10.0.0.2 y sin que le sea pedida contraseña al acceder.

Escribe los comandos openssh necesarios para preparar el acceso y abrir dicha sesión, indicando en que máquina y con que usuario se debe ejecutar cada comando.

Orden EDU/1406/2023, de 4 de diciembre (BOCyL de 12 de diciembre)

EJERCICIO 2.2 (0,5 Puntos)

Asumiendo que el fichero de definición de zona correspondiente al dominio *midominio.com* en un servidor DNS ya incluye un registro SOA, escribe las demás líneas necesarias para que el dominio disponga de:

- Un servidor DNS principal, llamado *ns1.midominio.com*, con IP 10.1.0.1
- Un servidor DNS secundario, llamado *ns2.midominio.com*, con IP 2701:fe9a:ed9e:1ee6
- Dos servidores de correo para el dominio: *mail1.midominio.com* (IP 10.1.0.2) y *mail2.midominio.com* (IP 10.1.0.3), con mayor prioridad para *mail2* que para *mail1*
- Una entrada DNS para *localhost*, con dirección IPv6 y nombre *h1.midominio.com*
- Un alias *h2.midominio.com*, que equivalga a la máquina que funciona como DNS principal

Orden EDU/1406/2023, de 4 de diciembre (BOCyL de 12 de diciembre)

BLOQUE III PROGRAMACIÓN (3 puntos)

(Este bloque se contestará en hojas separadas identificando claramente el número de bloque y ejercicio)

EJERCICIO 3.1 (0.6 Puntos)

Dado el siguiente código java:

```
public abstract class Insecto {  
    private String nombre;  
    private String familia;  
    Insecto(String n, String f){  
        nombre = n;  
        familia = f;  
    }  
}
```

- Declara un **interface** que se llame **Volador** que a su vez tenga 2 métodos, los cuales se llamaran **ponerHuevos** y **mudarAlas**.
- Hacer una clase **Mariposa** que herede de la clase Insecto, y que implemente la interfaz **Volador**.
- Hacer un constructor de la clase Mariposa que reciba como parámetros el nombre, la familia y el **número de huevos** que pone una mariposa. Añadir a la clase Mariposa los atributos y métodos que se consideren oportuno. Visualiza por pantalla en los métodos un mensaje explicativo de la acción que realiza, por ejemplo “poniendo”, “mudando”.
- Hacer un método sobrescrito **ponerHuevos** que reciba como parámetro el precio de venta de una mariposa, sabiendo que solo el 25% de los huevos llegan a desarrollarse. Visualízame por pantalla el ingreso que se prevé tener por ejemplar de mariposa.

Orden EDU/1406/2023, de 4 de diciembre (BOCyL de 12 de diciembre)

EJERCICIO 3.2 (0.4 Puntos)

Lea detenidamente el código de las 2 clases que se muestran a continuación y explicar cuál sería la salida por pantalla al ejecutar su código.

a)

```
public class Pregunta2a {
    public static void main(String[] args) {
        try {
            System.out.println("Paso 1");
            metodo();
            System.out.println("Paso 2");
        } catch (Exception e) {
            System.out.println("Paso 3");
        }
    }

    public static void metodo() throws Exception {
        try {
            System.out.println("Paso 4");
            try {
                System.out.println("Paso 5");
                Object o = null;
                o.toString();
                System.out.println("Paso 6");
            } catch (ArithmeticException aie) {
                System.out.println("Paso 7");
            }
            System.out.println("Paso 8");
        } catch (Exception e) {
            System.out.println("Paso 9");
            throw e;
        }
    }
}
//metodo
}
//Pregunta2a
```


Orden EDU/1406/2023, de 4 de diciembre (BOCyL de 12 de diciembre)

b)

```
public class Pregunta2b {

    public static double acceso_por_indice(double[] v, int j) throws RuntimeException{
        try{
            if((0<=j)&&(j<=v.length)) {
                return v[j];
            }//if
            else {

                throw new RuntimeException("El indice "+j+" no existe en el vector");

            }//else
        }//try
        catch (RuntimeException e)
        {
            System.out.println("ERROR en el valor del indice");
            throw e;
        }

    } //acceso_por_indice

    public static void main(String[] args) {

        double[]v=new double [15];
        acceso_por_indice (v,16);

    }
} // Pregunta2b
```

Orden EDU/1406/2023, de 4 de diciembre (BOCyL de 12 de diciembre)

EJERCICIO 3.3 (2 Puntos)

Encriptación: Método Bífido o de Polibio

La transmisión de mensajes a través de la red, requiere utilizar un Criptógrafo que cifre (transforme) los mensajes antes de enviarlos.

Polibio (Historiador Griego) está en el origen de un método muy original para cifrar (él describía un sistema de señales a distancia basado en el empleo de antorchas). Para esto, se disponen las letras en un cuadrado 5x5

En este ejemplo utilizaremos la siguiente matriz:

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I/J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Al ser una matriz de 5x5 **nos vemos obligados a cifrar de la misma forma la I y la J**. El contexto nos permitirá distinguir cual de las dos letras se pretendía cifrar.

Se reemplaza entonces cada letra por sus coordenadas en el cuadrado, escribiendo en primer lugar la fila y luego la columna. Por ejemplo, la A es reemplazada por 11, la B es reemplazado por 12, la F por 21, la M por 32.

Por ejemplo: **BUENOS DIAS genera** las siguientes coordenadas

12 45 15 33 34 43 14 24 11 43

y para proceder con la encriptación la cadena de coordenadas resultantes es dividida en dos partes iguales:

1245153334

4314241143

Si ahora leemos los números **como columnas** en lugar de por filas se forma una nueva cadena de coordenadas 14 23 41 54 12 54 31 31 34 43 las cuales formaran el nuevo mensaje encriptado **DHQYBY LLOS**.

Por tanto, debemos hacer un **programa en Java** que pueda ENCRIPtar una frase mediante el método Bífido. Cuando se encripte hay que visualizar por pantalla la frase encriptada con sus respectivos **espacios en blanco**.

Vamos a partir de la siguiente clase con una serie de atributos. Deberéis programar solo aquellos métodos que contengan en su interior asteriscos.

Orden EDU/1406/2023, de 4 de diciembre (BOCyL de 12 de diciembre)

No se pueden declarar más atributos de la clase EncriptarPolibio

Y además **TAMPOCO** se podrán redefinir los métodos que se han declarado en el enunciado, ni añadir alguno nuevo.

Clase principal desde la que se hacen las llamadas a la clase EncriptarPolibio. En esta clase NO hay que escribir nada de código.

```
public class Main {
    public static void main(String[] args) {

        EncriptarPolibio polibio1 = new EncriptarPolibio("HOLA MUNDO");
        polibio1.encriptarMensaje();
        polibio1.visualizarMensajeEncriptado();

    }
}
```

```
public class EncriptarPolibio {

    final char letras[ ][ ]={
        {'A','B','C','D','E'},
        {'F','G','H','J','K'},
        {'L','M','N','O','P'},
        {'Q','R','S','T','U'},
        {'V','W','X','Y','Z'}};

    String mensaje;
    int fila, columna;
    ArrayList<Integer> lista_encriptada = new ArrayList<>();

    public EncriptarPolibio(String palabra_encriptar) {
        this.mensaje = palabra_encriptar;
    }
}
```

Orden EDU/1406/2023, de 4 de diciembre (BOCyL de 12 de diciembre)

```
public void encriptarMensaje()  
  
//encriptara la frase que se ha pasado como atributo a la clase  
{  
  
    *****  
  
}  
  
public void visualizarMensajeEncriptado()  
// mostrara por pantalla el mensaje encriptado  
{  
  
    *****  
  
}  
  
}  
  
//visualizarMensajeEncriptado  
  
  
  
public void buscarCoordenada(char letra)  
//busca las coordenadas de la letra que se le pasa como parámetro  
{  
    *****  
  
}  
}
```

Orden EDU/1406/2023, de 4 de diciembre (BOCyL de 12 de diciembre)

BLOQUE IV BBDD (3 puntos)

(Este bloque se contestará en hojas separadas identificando claramente el número de bloque y ejercicio)

En una famosa comarca de Castilla y León se celebra todos los años en torno al verano una vuelta ciclista no profesional de cinco etapas. Participan media docena de equipos con un máximo de 10 ciclistas en cada uno.

Entre los objetivos de todos los ciclistas siempre está el poder llegar hasta la última etapa, aunque las caídas y otro tipo de incidentes provocan que algunos ciclistas tengan que abandonar antes de llegar a la meta.

En una base de datos, se dispone de las siguientes tablas:

EQUIPO

EQUIPO ID	NUMBER	NOT NULL
NOMBRE	VARCHAR2 (30)	NOT NULL
PAIS	VARCHAR2 (30)	NOT NULL
PRESUPUESTO	NUMBER	NOT NULL

EQUIPO_ID es PRIMARY KEY

ETAPA

ETAPA ID	NUMBER	NOT NULL
RECORRIDO	VARCHAR2 (30)	NOT NULL
FECHA	DATE	NOT NULL
KM	NUMBER	NOT NULL

ETAPA_ID es PRIMARY KEY

CICLISTA

CICLISTA ID	NUMBER	NOT NULL
NOMBRE	VARCHAR2 (30)	NOT NULL
EDAD	NUMBER	NOT NULL
NACIONALIDAD	VARCHAR2 (30)	NOT NULL
ALTURA	NUMBER	NOT NULL
PESO	NUMBER	NOT NULL
EQUIPO ID	NUMBER	NOT NULL

CICLISTA_ID es PRIMARY KEY

EQUIPO_ID es FOREIGN KEY a la tabla EQUIPO

PARTICIPACION

CICLISTA ID	NUMBER	NOT NULL
ETAPA ID	NUMBER	NOT NULL
PUESTO	NUMBER	NOT NULL
TIEMPO	NUMBER	NOT NULL

CICLISTA_ID, ETAPA_ID es PRIMARY KEY

CICLISTA_ID es FOREIGN KEY a la tabla CICLISTA

ETAPA_ID es FOREIGN KEY a la table ETAPA

Realizar en Oracle las siguientes tareas:

EJERCICIO 4.1 (1.5 Puntos)

Mostrar mediante una sola consulta SQL el nombre de los ciclistas que, al menos en dos etapas hayan conseguido un puesto 1, 2 o 3. En la consulta debe aparecer ordenada por puesto la siguiente información: nombre, puesto y número de veces que han conseguido dicho puesto.

Mostrar mediante una sola consulta SQL, el nombre y la edad de los ciclistas, junto a un DATO que indicará:

- '>PESO' cuando el peso del ciclista supere la media del peso de los ciclistas de su equipo.
- '<ALTURA' cuando la altura del ciclista sea inferior a media de la altura de los ciclistas de su misma edad.

Orden EDU/1406/2023, de 4 de diciembre (BOCyL de 12 de diciembre)

Teniendo en cuenta que cuando un ciclista no finaliza una etapa se le pone un 0 en el campo PUESTO de la tabla PARTICIPACION, crear una función que reciba como parámetro un identificador de ciclista y devuelva la suma de kms de las etapas que ha finalizado, siempre que no sea la última en la que ha participado. (Es obligatorio crear también el bloque PL/SQL para probar la función).

EJERCICIO 4.2 (1.5 Puntos)

Crear la siguiente tabla:

CLASIFICACION

CICLISTA_ID	NUMBER	NOT NULL
PUNTOS	NUMBER	NOT NULL

CICLISTA_ID es PRIMARY KEY

Insertar en la tabla CLASIFICACION tantos registros como ciclistas participan en la vuelta.

Crear un bloque para actualizar el campo PUNTOS de la tabla anterior, con los puntos que les corresponden a los ciclistas según los puestos obtenidos en las etapas en las que han participado (tabla PARTICIPACION).

En función de la posición lograda en cada etapa, obtendrán los siguientes puntos:

- 1ª posición: 30 puntos.
- 2ª posición: 20 puntos.
- 3ª posición: 10 puntos.

Crear una tabla llamada AUDITORIA con una única columna con nombre INFO tipo VARCHAR2(100).

Crear un disparador llamado D_CLASIFICACION para auditar la inserción, modificación y borrado en la tabla CLASIFICACION, insertando un mensaje en la tabla AUDITORIA en función de la operación realizada.

- Si se inserta, el mensaje incluirá: INSERCIÓN, fecha, identificador del ciclista y puntos.
- Si se modifica, el mensaje incluirá MODIFICACIÓN, fecha, nombre e identificación del ciclista, puntos anteriores y puntos finales.
- Si se borra, el mensaje incluirá: BORRADO, fecha, identificador del ciclista y puntos.